
Ironic Documentation

Release 18.0.1.dev14

OpenStack Foundation

Aug 25, 2021

CONTENTS

1	Introduction	1
2	Installation Guide	3
2.1	Bare Metal Service Installation Guide	3
2.2	Bare Metal Service Upgrade Guide	105
3	User Guide	117
3.1	Bare Metal Service User Guide	117
4	Administrator Guide	131
4.1	Drivers, Hardware Types and Hardware Interfaces	131
4.2	Administrators Guide	229
5	Configuration Guide	355
5.1	Configuration Reference	355
6	Bare Metal API References	523
6.1	REST API Conceptual Guide	523
7	Command References	537
7.1	Command References	537
8	Contributor Guide	543
8.1	Developers Guide	543
9	Release Notes	3421
	Python Module Index	3423
	Index	3427

INTRODUCTION

Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines. It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance), and Object (swift) services.

The Bare Metal service manages hardware through both common (eg. PXE and IPMI) and vendor-specific remote management protocols. It provides the cloud operator with a unified interface to a heterogeneous fleet of servers while also providing the Compute service with an interface that allows physical servers to be managed as though they were virtual machines.

This documentation is continually updated and may not represent the state of the project at any specific prior release. To access documentation for a previous release of ironic, append the OpenStack release name to the URL; for example, the ocat release is available at <https://docs.openstack.org/ironic/ocata/>.

Found a bug in one of our projects? Please see *Bug Reporting and Triaging Guide*.

INSTALLATION GUIDE

2.1 Bare Metal Service Installation Guide

The Bare Metal service is a collection of components that provides support to manage and provision physical machines.

This chapter assumes a working setup of OpenStack following the [OpenStack Installation Guides](#). It contains the following sections:

2.1.1 Bare Metal service overview

The Bare Metal service, codenamed `ironic`, is a collection of components that provides support to manage and provision physical machines.

Bare Metal service components

The Bare Metal service includes the following components:

ironic-api A RESTful API that processes application requests by sending them to the `ironic-conductor` over [remote procedure call \(RPC\)](#). Can be run through [WSGI](#) or as a separate process.

ironic-conductor Adds/edits/deletes nodes; powers on/off nodes with IPMI or other vendor-specific protocol; provisions/deploys/cleans bare metal nodes.

`ironic-conductor` uses *drivers* to execute operations on hardware.

ironic-python-agent A python service which is run in a temporary ramdisk to provide `ironic-conductor` and `ironic-inspector` services with remote access, in-band hardware control, and hardware introspection.

Additionally, the Bare Metal service has certain external dependencies, which are very similar to other OpenStack services:

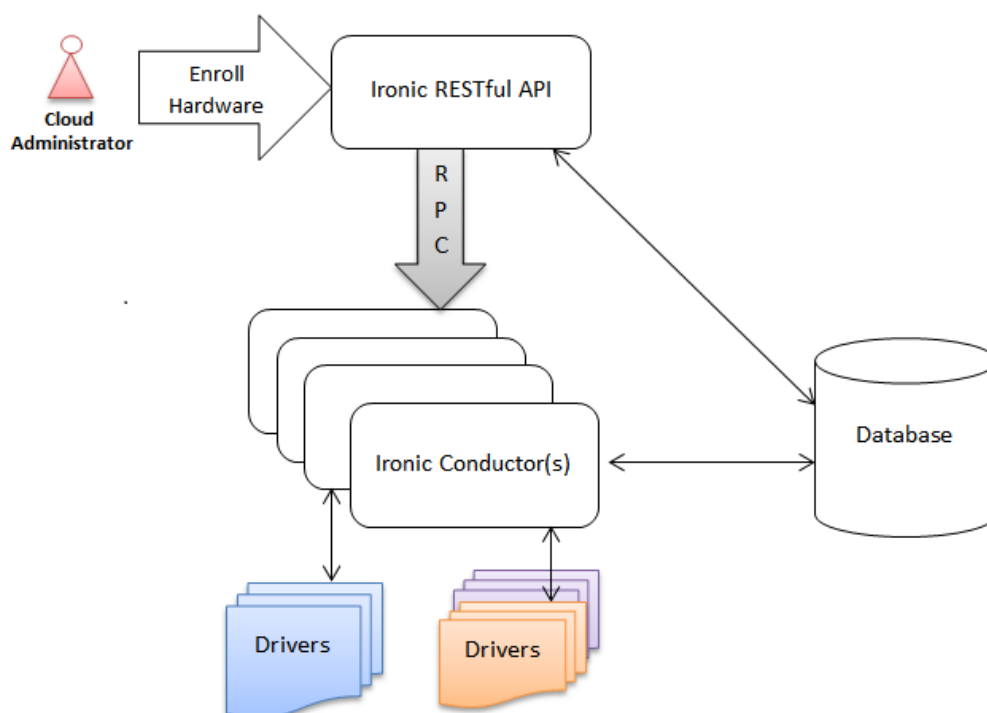
- A database to store hardware information and state. You can set the database back-end type and location. A simple approach is to use the same database back end as the Compute service. Another approach is to use a separate database back-end to further isolate bare metal resources (and associated metadata) from users.
- An [oslo.messaging](#) compatible queue, such as RabbitMQ. It may use the same implementation as that of the Compute service, but that is not a requirement. Used to implement RPC between `ironic-api` and `ironic-conductor`.

Deployment architecture

The Bare Metal RESTful API service is used to enroll hardware that the Bare Metal service will manage. A cloud administrator usually registers it, specifying their attributes such as MAC addresses and IPMI credentials. There can be multiple instances of the API service.

The *ironic-conductor* process does the bulk of the work. For security reasons, it is advisable to place it on an isolated host, since it is the only service that requires access to both the data plane and IPMI control plane.

There can be multiple instances of the conductor service to support various class of drivers and also to manage fail over. Instances of the conductor service should be on separate nodes. Each conductor can itself run many drivers to operate heterogeneous hardware. This is depicted in the following figure.



The API exposes a list of supported drivers and the names of conductor hosts servicing them.

Interaction with OpenStack components

The Bare Metal service may, depending upon configuration, interact with several other OpenStack services. This includes:

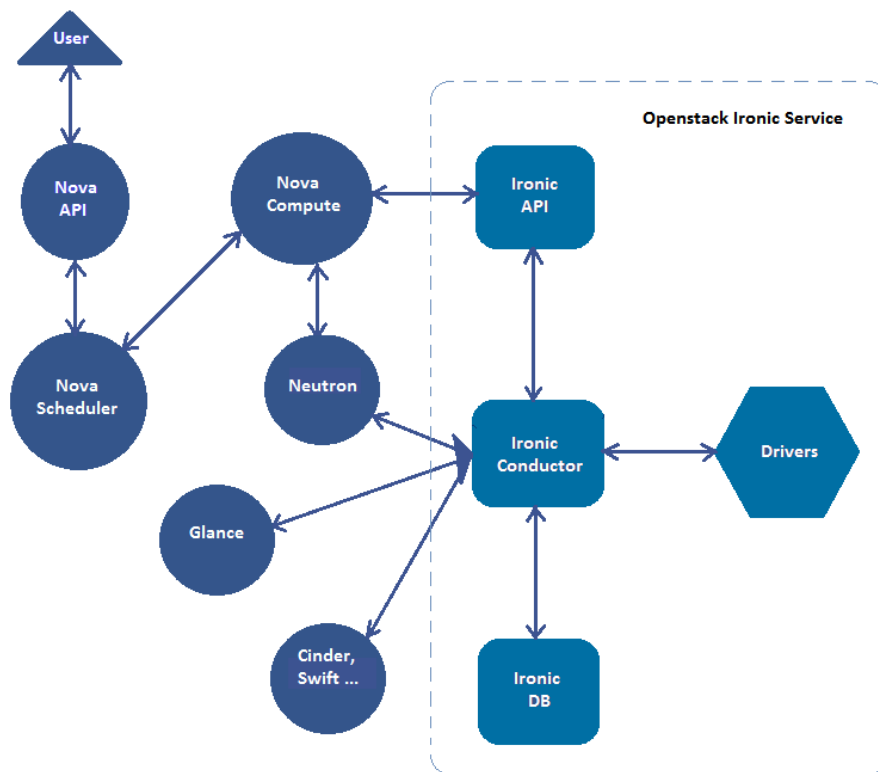
- the OpenStack Telemetry module (`ceilometer`) for consuming the IPMI metrics
- the OpenStack Identity service (`keystone`) for request authentication and to locate other OpenStack services
- the OpenStack Image service (`glance`) from which to retrieve images and image meta-data
- the OpenStack Networking service (`neutron`) for DHCP and network configuration

- the OpenStack Compute service (nova) works with the Bare Metal service and acts as a user-facing API for instance management, while the Bare Metal service provides the admin/operator API for hardware management. The OpenStack Compute service also provides scheduling facilities (matching flavors <-> images <-> hardware), tenant quotas, IP assignment, and other services which the Bare Metal service does not, in and of itself, provide.
- the OpenStack Object Storage (swift) provides temporary storage for the configdrive, user images, deployment logs and inspection data.

Logical architecture

The diagram below shows the logical architecture. It shows the basic components that form the Bare Metal service, the relation of the Bare Metal service with other OpenStack services and the logical flow of a boot instance request resulting in the provisioning of a physical server.

Figure 1.2. Logical Architecture



A user's request to boot an instance is passed to the Compute service via the Compute API and the Compute Scheduler. The Compute service uses the *ironic virt driver* to hand over this request to the Bare Metal service, where the request passes from the Bare Metal API, to the Conductor, to a Driver to successfully provision a physical server for the user.

Just as the Compute service talks to various OpenStack services like Image, Network, Object Store etc to provision a virtual machine instance, here the Bare Metal service talks to the same OpenStack services for image, network and other resource needs to provision a bare metal instance.

See *Understanding Bare Metal Deployment* for a more detailed breakdown of a typical deployment process.

Associated projects

Optionally, one may wish to utilize the following associated projects for additional functionality:

python-ironicclient A command-line interface (CLI) and python bindings for interacting with the Bare Metal service.

ironic-ui Horizon dashboard, providing graphical interface (GUI) for the Bare Metal API.

ironic-inspector An associated service which performs in-band hardware introspection by PXE booting unregistered hardware into the ironic-python-agent ramdisk.

diskimage-builder A related project to help facilitate the creation of ramdisks and machine images, such as those running the ironic-python-agent.

bifrost A set of Ansible playbooks that automates the task of deploying a base image onto a set of known hardware using ironic in a standalone mode.

2.1.2 Reference Deploy Architectures

This section covers the way we recommend the Bare Metal service to be deployed and managed. It is assumed that a reader has already gone through *Bare Metal Service User Guide*. It may be also useful to try *Deploying IroniC with DevStack* first to get better familiar with the concepts used in this guide.

Common Considerations

This section covers considerations that are equally important to all described architectures.

- *Components*
- *Hardware and drivers*
 - *Power and management interfaces*
 - *Boot interface*
 - *Hardware specifications*
- *Image types*
- *Local vs network boot*
- *Networking*
- *HA and Scalability*
 - *ironic-api*
 - *ironic-conductor*
 - * *High availability*
 - * *Performance*
 - * *Disk space*
 - *Other services*

Components

As explained in [Bare Metal service overview](#), the Bare Metal service has three components.

- The Bare Metal API service (`ironic-api`) should be deployed in a similar way as the control plane API services. The exact location will depend on the architecture used.
- The Bare Metal conductor service (`ironic-conductor`) is where most of the provisioning logic lives. The following considerations are the most important when deciding on the way to deploy it:
 - The conductor manages a certain proportion of nodes, distributed to it via a hash ring. This includes constantly polling these nodes for their current power state and hardware sensor data (if enabled and supported by hardware, see [Collecting sensor data](#) for an example).
 - The conductor needs access to the [management controller](#) of each node it manages.
 - The conductor co-exists with TFTP (for PXE) and/or HTTP (for iPXE) services that provide the kernel and ramdisk to boot the nodes. The conductor manages them by writing files to their root directories.
 - If serial console is used, the conductor launches console processes locally. If the `nova-serialproxy` service (part of the Compute service) is used, it has to be able to reach the conductors. Otherwise, they have to be directly accessible by the users.
 - There must be mutual connectivity between the conductor and the nodes being deployed or cleaned. See [Networking](#) for details.
- The provisioning ramdisk which runs the `ironic-python-agent` service on start up.

Warning: The `ironic-python-agent` service is not intended to be used or executed anywhere other than a provisioning/cleaning/rescue ramdisk.

Hardware and drivers

The Bare Metal service strives to provide the best support possible for a variety of hardware. However, not all hardware is supported equally well. It depends on both the capabilities of hardware itself and the available drivers. This section covers various considerations related to the hardware interfaces. See [Enabling drivers and hardware types](#) for a detailed introduction into hardware types and interfaces before proceeding.

Power and management interfaces

The minimum set of capabilities that the hardware has to provide and the driver has to support is as follows:

1. getting and setting the power state of the machine
2. getting and setting the current boot device
3. booting an image provided by the Bare Metal service (in the simplest case, support booting using [PXE](#) and/or [iPXE](#))

Note: Strictly speaking, it is possible to make the Bare Metal service provision nodes without some of these capabilities via some manual steps. It is not the recommended way of deployment, and thus it is not covered in this guide.

Once you make sure that the hardware supports these capabilities, you need to find a suitable driver. Most of enterprise-grade hardware has support for IPMI and thus can utilize *IPMI driver*. Some newer hardware also supports *Redfish driver*. Several vendors provide more specific drivers that usually provide additional capabilities. Check *Drivers, Hardware Types and Hardware Interfaces* to find the most suitable one.

Boot interface

The boot interface of a node manages booting of both the deploy ramdisk and the user instances on the bare metal node. The deploy interface orchestrates the deployment and defines how the image gets transferred to the target disk.

The main alternatives are to use PXE/iPXE or virtual media - see *Boot interfaces* for a detailed explanation. If a virtual media implementation is available for the hardware, it is recommended using it for better scalability and security. Otherwise, it is recommended to use iPXE, when it is supported by target hardware.

Hardware specifications

The Bare Metal services does not impose too many restrictions on the characteristics of hardware itself. However, keep in mind that

- By default, the Bare Metal service will pick the smallest hard drive that is larger than 4 GiB for deployment. Another hard drive can be used, but it requires setting *root device hints*.

Note: This device does not have to match the boot device set in BIOS (or similar firmware).

- The machines should have enough RAM to fit the deployment/cleaning ramdisk to run. The minimum varies greatly depending on the way the ramdisk was built. For example, *tinyipa*, the TinyCoreLinux-based ramdisk used in the CI, only needs 400 MiB of RAM, while ramdisks built by *diskimage-builder* may require 3 GiB or more.

Image types

The Bare Metal service can deploy two types of images:

- *Whole-disk* images that contain a complete partitioning table with all necessary partitions and a bootloader. Such images are the most universal, but may be harder to build.
- *Partition images* that contain only the root partition. The Bare Metal service will create the necessary partitions and install a boot loader, if needed.

Warning: Partition images are only supported with GNU/Linux operating systems.

Warning: If you plan on using local boot, your partition images must contain GRUB2 boot-loader tools to enable ironic to set up the bootloader during deploy.

Local vs network boot

The Bare Metal service supports booting user instances either using a local bootloader or using the drivers boot interface (e.g. via [PXE](#) or [iPXE](#) protocol in case of the `pxe` interface).

Network boot cannot be used with certain architectures (for example, when no tenant networks have access to the control plane).

Additional considerations are related to the `pxe` boot interface, and other boot interfaces based on it:

- Local boot makes nodes boot process independent of the Bare Metal conductor managing it. Thus, nodes are able to reboot correctly, even if the Bare Metal TFTP or HTTP service is down.
- Network boot (and iPXE) must be used when booting nodes from remote volumes, if the driver does not support attaching volumes out-of-band.

The default boot option for the cloud can be changed via the Bare Metal service configuration file, for example:

```
[deploy]
default_boot_option = local
```

This default can be overridden by setting the `boot_option` capability on a node. See [Local boot with partition images](#) for details.

Note: Currently, local boot is used by default. Its safer to set the `default_boot_option` explicitly.

Networking

There are several recommended network topologies to be used with the Bare Metal service. They are explained in depth in specific architecture documentation. However, several considerations are common for all of them:

- There has to be a *provisioning* network, which is used by nodes during the deployment process. If allowed by the architecture, this network should not be accessible by end users, and should not have access to the internet.
- There has to be a *cleaning* network, which is used by nodes during the cleaning process.
- There should be a *rescuing* network, which is used by nodes during the rescue process. It can be skipped if the rescue process is not supported.

Note: In the majority of cases, the same network should be used for cleaning, provisioning and rescue for simplicity.

Unless noted otherwise, everything in these sections apply to all three networks.

- The baremetal nodes must have access to the Bare Metal API while connected to the provisioning/cleaning/rescuing network.

Note: Only two endpoints need to be exposed there:

```
GET /v1/lookup
POST /v1/heartbeat/[a-z0-9\-_]+
```

You may want to limit access from this network to only these endpoints, and make these endpoint not accessible from other networks.

- If the pxe boot interface (or any boot interface based on it) is used, then the baremetal nodes should have untagged (access mode) connectivity to the provisioning/cleaning/rescuing networks. It allows PXE firmware, which does not support VLANs, to communicate with the services required for provisioning.

Note: It depends on the *network interface* whether the Bare Metal service will handle it automatically. Check the networking documentation for the specific architecture.

Sometimes it may be necessary to disable the spanning tree protocol delay on the switch - see *DHCP during PXE or iPXE is inconsistent or unreliable*.

- The Baremetal nodes need to have access to any services required for provisioning/cleaning/rescue, while connected to the provisioning/cleaning/rescuing network. This may include:
 - a TFTP server for PXE boot and also an HTTP server when iPXE is enabled
 - either an HTTP server or the Object Storage service in case of the direct deploy interface and some virtual media boot interfaces
- The Baremetal Conductors need to have access to the booted baremetal nodes during provisioning/cleaning/rescue. A conductor communicates with an internal API, provided by **ironic-python-agent**, to conduct actions on nodes.

HA and Scalability

ironic-api

The Bare Metal API service is stateless, and thus can be easily scaled horizontally. It is recommended to deploy it as a WSGI application behind e.g. Apache or another WSGI container.

Note: This service accesses the ironic database for reading entities (e.g. in response to GET /v1/nodes request) and in rare cases for writing.

ironic-conductor

High availability

The Bare Metal conductor service utilizes the active/active HA model. Every conductor manages a certain subset of nodes. The nodes are organized in a hash ring that tries to keep the load spread more or less uniformly across the conductors. When a conductor is considered offline, its nodes are taken over by other conductors. As a result of this, you need at least 2 conductor hosts for an HA deployment.

Performance

Conductors can be resource intensive, so it is recommended (but not required) to keep all conductors separate from other services in the cloud. The minimum required number of conductors in a deployment depends on several factors:

- the performance of the hardware where the conductors will be running,
- the speed and reliability of the `management controller` of the bare metal nodes (for example, handling slower controllers may require having less nodes per conductor),
- the frequency, at which the management controllers are polled by the Bare Metal service (see the `sync_power_state_interval` option),
- the bare metal driver used for nodes (see *Hardware and drivers* above),
- the network performance,
- the maximum number of bare metal nodes that are provisioned simultaneously (see the `max_concurrent_builds` option for the Compute service).

We recommend a target of **100** bare metal nodes per conductor for maximum reliability and performance. There is some tolerance for a larger number per conductor. However, it was reported¹² that reliability degrades when handling approximately 300 bare metal nodes per conductor.

Disk space

Each conductor needs enough free disk space to cache images it uses. Depending on the combination of the deploy interface and the boot option, the space requirements are different:

- The deployment kernel and ramdisk are always cached during the deployment.
- When `[agent] image_download_source` is set to `http` and Glance is used, the conductor will download instances images locally to serve them from its HTTP server. Use `swift` to publish images using temporary URLs and convert them on the nodes side.

When `[agent] image_download_source` is set to `local`, it will happen even for HTTP(s) URLs. For standalone case use `http` to avoid unnecessary caching of images.

In both cases a cached image is converted to raw if `force_raw_images` is `True` (the default).

¹ <http://lists.openstack.org/pipermail/openstack-dev/2017-June/118033.html>

² <http://lists.openstack.org/pipermail/openstack-dev/2017-June/118327.html>

Note: `image_download_source` can also be provided in the nodes `driver_info` or `instance_info`. See *Deploy with custom HTTP servers*.

- When network boot is used, the instance image kernel and ramdisk are cached locally while the instance is active.

Note: All images may be stored for some time after they are no longer needed. This is done to speed up simultaneous deployments of many similar images. The caching can be configured via the `image_cache_size` and `image_cache_ttl` configuration options in the `pxe` group.

Other services

When integrating with other OpenStack services, more considerations may need to be applied. This is covered in other parts of this guide.

Scenarios

Small cloud with trusted tenants

Story

As an operator I would like to build a small cloud with both virtual and bare metal instances or add bare metal provisioning to my existing small or medium scale single-site OpenStack cloud. The expected number of bare metal machines is less than 100, and the rate of provisioning and unprovisioning is expected to be low. All users of my cloud are trusted by me to not conduct malicious actions towards each other or the cloud infrastructure itself.

As a user I would like to occasionally provision bare metal instances through the Compute API by selecting an appropriate Compute flavor. I would like to be able to boot them from images provided by the Image service or from volumes provided by the Volume service.

Components

This architecture assumes an [OpenStack installation](#) with the following components participating in the bare metal provisioning:

- The [Compute service](#) manages bare metal instances.
- The [Networking service](#) provides DHCP for bare metal instances.
- The [Image service](#) provides images for bare metal instances.

The following services can be optionally used by the Bare Metal service:

- The [Volume service](#) provides volumes to boot bare metal instances from.
- The [Bare Metal Introspection service](#) simplifies enrolling new bare metal machines by conducting in-band introspection.

Node roles

An OpenStack installation in this guide has at least these three types of nodes:

- A *controller* node hosts the control plane services.
- A *compute* node runs the virtual machines and hosts a subset of Compute and Networking components.
- A *block storage* node provides persistent storage space for both virtual and bare metal nodes.

The *compute* and *block storage* nodes are configured as described in the installation guides of the [Compute service](#) and the [Volume service](#) respectively. The *controller* nodes host the Bare Metal service components.

Networking

The networking architecture will highly depend on the exact operating requirements. This guide expects the following existing networks: *control plane*, *storage* and *public*. Additionally, two more networks will be needed specifically for bare metal provisioning: *bare metal* and *management*.

Control plane network

The *control plane network* is the network where OpenStack control plane services provide their public API.

The Bare Metal API will be served to the operators and to the Compute service through this network.

Public network

The *public network* is used in a typical OpenStack deployment to create floating IPs for outside access to instances. Its role is the same for a bare metal deployment.

Note: Since, as explained below, bare metal nodes will be put on a flat provider network, it is also possible to organize direct access to them, without using floating IPs and bypassing the Networking service completely.

Bare metal network

The *Bare metal network* is a dedicated network for bare metal nodes managed by the Bare Metal service.

This architecture uses *flat bare metal networking*, in which both tenant traffic and technical traffic related to the Bare Metal service operation flow through this one network. Specifically, this network will serve as the *provisioning*, *cleaning* and *rescuing* network. It will also be used for introspection via the Bare Metal Introspection service. See [common networking considerations](#) for an in-depth explanation of the networks used by the Bare Metal service.

DHCP and boot parameters will be provided on this network by the Networking services DHCP agents.

For booting from volumes this network has to have a route to the *storage network*.

Management network

Management network is an independent network on which BMCs of the bare metal nodes are located.

The `ironic-conductor` process needs access to this network. The tenants of the bare metal nodes must not have access to it.

Note: The *direct deploy interface* and certain *Drivers, Hardware Types and Hardware Interfaces* require the *management network* to have access to the Object storage service backend.

Controllers

A *controller* hosts the OpenStack control plane services as described in the [control plane design guide](#). While this architecture allows using *controllers* in a non-HA configuration, it is recommended to have at least three of them for HA. See [HA and Scalability](#) for more details.

Bare Metal services

The following components of the Bare Metal service are installed on a *controller* (see [components of the Bare Metal service](#)):

- The Bare Metal API service either as a WSGI application or the `ironic-api` process. Typically, a load balancer, such as HAProxy, spreads the load between the API instances on the *controllers*.

The API has to be served on the *control plane network*. Additionally, it has to be exposed to the *bare metal network* for the ramdisk callback API.

- The `ironic-conductor` process. These processes work in active/active HA mode as explained in [HA and Scalability](#), thus they can be installed on all *controllers*. Each will handle a subset of bare metal nodes.

The `ironic-conductor` processes have to have access to the following networks:

- *control plane* for interacting with other services
 - *management* for contacting nodes BMCs
 - *bare metal* for contacting deployment, cleaning or rescue ramdisks
- TFTP and HTTP service for booting the nodes. Each `ironic-conductor` process has to have a matching TFTP and HTTP service. They should be exposed only to the *bare metal network* and must not be behind a load balancer.
 - The `nova-compute` process (from the Compute service). These processes work in active/active HA mode when dealing with bare metal nodes, thus they can be installed on all *controllers*. Each will handle a subset of bare metal nodes.

Note: There is no 1-1 mapping between `ironic-conductor` and `nova-compute` processes, as they communicate only through the Bare Metal API service.

- The `networking-baremetal` ML2 plugin should be loaded into the Networking service to assist with binding bare metal ports.

The `ironic-neutron-agent` service should be started as well.

- If the Bare Metal introspection is used, its `ironic-inspector` process has to be installed on all *controllers*. Each such process works as both Bare Metal Introspection API and conductor service. A load balancer should be used to spread the API load between *controllers*.

The API has to be served on the *control plane network*. Additionally, it has to be exposed to the *bare metal network* for the ramdisk callback API.

Shared services

A *controller* also hosts two services required for the normal operation of OpenStack:

- Database service (MySQL/MariaDB is typically used, but other enterprise-grade database solutions can be used as well).

All Bare Metal service components need access to the database service.

- Message queue service (RabbitMQ is typically used, but other enterprise-grade message queue brokers can be used as well).

Both Bare Metal API (WSGI application or `ironic-api` process) and the `ironic-conductor` processes need access to the message queue service. The Bare Metal Introspection service does not need it.

Note: These services are required for all OpenStack services. If you're adding the Bare Metal service to your cloud, you may reuse the existing database and messaging queue services.

Bare metal nodes

Each bare metal node must be capable of booting from network, virtual media or other boot technology supported by the Bare Metal service as explained in *Boot interface*. Each node must have one NIC on the *bare metal network*, and this NIC (and **only** it) must be configured to be able to boot from network. This is usually done in the *BIOS setup* or a similar firmware configuration utility. There is no need to alter the boot order, as it is managed by the Bare Metal service. Other NICs, if present, will not be managed by OpenStack.

The NIC on the *bare metal network* should have untagged connectivity to it, since PXE firmware usually does not support VLANs - see *Networking* for details.

Storage

If your hardware **and** its bare metal *driver* support booting from remote volumes, please check the driver documentation for information on how to enable it. It may include routing *management* and/or *bare metal* networks to the *storage network*.

In case of the standard *PXE boot*, booting from remote volumes is done via iPXE. In that case, the Volume storage backend must support iSCSI protocol, and the *bare metal network* has to have a route to the *storage network*. See *Boot From Volume* for more details.

2.1.3 Install and configure the Bare Metal service

This section describes how to install and configure the Bare Metal service, code-named ironic.

Note that installation and configuration vary by distribution.

Install and configure for Red Hat Enterprise Linux and CentOS

This section describes how to install and configure the Bare Metal service for Red Hat Enterprise Linux 8 and CentOS 8.

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages (using dnf)

```
# dnf install openstack-ironic-api openstack-ironic-conductor python3-
↪ironicclient
```

2. Enable services

```
# systemctl enable openstack-ironic-api openstack-ironic-conductor
# systemctl start openstack-ironic-api openstack-ironic-conductor
```

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an ironic-api and an ironic-conductor services on the same host or different hosts. Users also can add new ironic-conductor hosts to deal with an increasing number of bare metal nodes. But the additional ironic-conductor services should be at the same version as that of existing ironic-conductor services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↪charset=utf8
```

2. Configure the ironic-api service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authentication:

```

[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```

[json_rpc]
port = 9999

```

3. Configure the ironic-api service to use these credentials with the Identity service. Replace PUBLIC_IDENTITY_IP with the public IP of the Identity server, PRIVATE_IDENTITY_IP with the private IP of the Identity server and replace IRONIC_PASSWORD with the password you chose for the ironic user in the Identity service:

```

[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

```

(continues on next page)

(continued from previous page)

```
# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

4. Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

5. Restart the ironic-api service:

Fedora/RHEL8/CentOS8/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

1. Install the apache service:

Fedora/RHEL8/CentOS8:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

2. Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL8/CentOS8:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

3. Edit the recently copied `<apache-configuration-dir>/ironic.conf`:
 1. Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
 2. Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
 3. Modify the `Directory` directive to set the path to the Ironic API code.
 4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL8/CentOS8:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:

```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by `pbr` and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for `gunicorn`:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \
    'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-file=/
    ↪path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]

# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. Ironic-conductor should use the same configuration as `ironic-api`. Replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↪charset=utf8
```

3. Configure the `ironic-conductor` service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as `ironic-api`. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]
host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- [neutron] - to access the OpenStack Networking service

- `[glance]` - to access the OpenStack Image service
- `[swift]` - to access the OpenStack Object Storage service
- `[cinder]` - to access the OpenStack Block Storage service
- `[inspector]` - to access the OpenStack Bare Metal Introspection service
- `[service_catalog]` - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the `[keystone_authtoken]` section for the `ironic-api` service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with `Authentication plugin`, `Session` and `Adapter` concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as `Session`- and `Adapter`-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for `ironic-api` service
- dynamic password authentication plugin that will discover appropriate version of Identity service API based on other provided options
 - replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default
```

(continues on next page)

(continued from previous page)

```
# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.
6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

Install and configure for Ubuntu

This section describes how to install and configure the Bare Metal service for Ubuntu 14.04 (LTS).

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages (using `apt-get`)

```
# apt-get install ironic-api ironic-conductor python3-ironicclient
```

2. Enable services

Services are enabled by default on Ubuntu.

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an `ironic-api` and an `ironic-conductor` services on the same host or different hosts. Users also can add new `ironic-conductor` hosts to deal with an increasing number of bare metal nodes. But the additional `ironic-conductor` services should be at the same version as that of existing `ironic-conductor` services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

2. Configure the `ironic-api` service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between `ironic-conductor` and `ironic-api`. Enable it in the configuration and provide the keystone credentials to use for authentication:

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default
```

(continues on next page)

(continued from previous page)

```
# User's domain id (string value)
user_domain_id=default
```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```
[json_rpc]
port = 9999
```

3. Configure the ironic-api service to use these credentials with the Identity service. Replace PUBLIC_IDENTITY_IP with the public IP of the Identity server, PRIVATE_IDENTITY_IP with the private IP of the Identity server and replace IRONIC_PASSWORD with the password you chose for the ironic user in the Identity service:

```
[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

4. Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

5. Restart the ironic-api service:

Fedora/RHEL8/CentOS8/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

1. Install the apache service:

Fedora/RHEL8/CentOS8:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

2. Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL8/CentOS8:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

3. Edit the recently copied `<apache-configuration-dir>/ironic.conf`:

1. Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
2. Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
3. Modify the `Directory` directive to set the path to the Ironic API code.

4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL8/CentOS8:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:

```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by `pbr` and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for `gunicorn`:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \
  'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-file=/
  ↪path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]
```

```
# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. Ironic-conductor should use the same configuration as ironic-api. Replace IRONIC_DBPASSWORD with the password of your ironic user, and replace DB_IP with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

3. Configure the ironic-conductor service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as ironic-api. Replace RPC_* with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default
```

(continues on next page)

(continued from previous page)

```
# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]
host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- `[neutron]` - to access the OpenStack Networking service
- `[glance]` - to access the OpenStack Image service
- `[swift]` - to access the OpenStack Object Storage service
- `[cinder]` - to access the OpenStack Block Storage service
- `[inspector]` - to access the OpenStack Bare Metal Introspection service
- `[service_catalog]` - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the `[keystone_authtoken]` section for the ironic-api service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with `Authentication plugin`, `Session` and `Adapter` concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as `Session`- and `Adapter`-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for ironic-api service
- dynamic password authentication plugin that will discover appropriate version of Identity service API based on other provided options

- replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.
6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

Install and configure for openSUSE and SUSE Linux Enterprise

This section describes how to install and configure the Bare Metal service for openSUSE Leap 42.2 and SUSE Linux Enterprise Server 12 SP2.

Note: Installation of the Bare Metal service on openSUSE and SUSE Linux Enterprise Server is not officially supported. Nevertheless, installation should be possible.

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages

```
# zypper install openstack-ironic-api openstack-ironic-conductor python3-ironicclient
```

2. Enable services

```
# systemctl enable openstack-ironic-api openstack-ironic-conductor
# systemctl start openstack-ironic-api openstack-ironic-conductor
```

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an ironic-api and an ironic-conductor services on the same host or different hosts. Users also can add new ironic-conductor hosts to deal with an increasing number of bare metal nodes. But the additional ironic-conductor services should be at the same version as that of existing ironic-conductor services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

2. Configure the ironic-api service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authentication:

```

[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```

[json_rpc]
port = 9999

```

3. Configure the ironic-api service to use these credentials with the Identity service. Replace PUBLIC_IDENTITY_IP with the public IP of the Identity server, PRIVATE_IDENTITY_IP with the private IP of the Identity server and replace IRONIC_PASSWORD with the password you chose for the ironic user in the Identity service:

```

[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

```

(continues on next page)

(continued from previous page)

```
# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

4. Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

5. Restart the ironic-api service:

Fedora/RHEL8/CentOS8/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

1. Install the apache service:

Fedora/RHEL8/CentOS8:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:


```
zypper install apache2
```

2. Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL8/CentOS8:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

3. Edit the recently copied `<apache-configuration-dir>/ironic.conf`:
 1. Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
 2. Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
 3. Modify the `Directory` directive to set the path to the Ironic API code.
 4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL8/CentOS8:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:

```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by `pbr` and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for `gunicorn`:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \  
  'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-file=/  
↪path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]  
  
# IP address of this host. If unset, will determine the IP  
# programmatically. If unable to do so, will use "127.0.0.1".  
# (string value)  
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. Ironic-conductor should use the same configuration as `ironic-api`. Replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]  
  
# The SQLAlchemy connection string to use to connect to the  
# database. (string value)  
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?  
↪charset=utf8
```

3. Configure the `ironic-conductor` service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as `ironic-api`. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]  
  
# A URL representing the messaging driver to use and its full  
# configuration. (string value)  
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]
host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- [neutron] - to access the OpenStack Networking service

- `[glance]` - to access the OpenStack Image service
- `[swift]` - to access the OpenStack Object Storage service
- `[cinder]` - to access the OpenStack Block Storage service
- `[inspector]` - to access the OpenStack Bare Metal Introspection service
- `[service_catalog]` - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the `[keystone_authtoken]` section for the `ironic-api` service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with `Authentication plugin`, `Session` and `Adapter` concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as `Session`- and `Adapter`-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for `ironic-api` service
- dynamic password authentication plugin that will discover appropriate version of Identity service API based on other provided options
 - replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default
```

(continues on next page)

(continued from previous page)

```
# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.
6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

2.1.4 Building or downloading a deploy ramdisk image

Ironic depends on having an image with the `ironic-python-agent` (IPA) service running on it for controlling and deploying bare metal nodes.

Two kinds of images are published on every commit from every branch of `ironic-python-agent` (IPA)

- **DIB** images are suitable for production usage and can be downloaded from <https://tarballs.openstack.org/ironic-python-agent/dib/files/>.
 - For Train and older use CentOS 7 images.
 - For Ussuri and newer use CentOS 8 images.

Warning: CentOS 7 master images are no longer updated and must not be used.

Warning: The published images will not work for dhcp-less deployments since the `simple-init` element is not present. Check the **DIB** documentation to see how to build the image.

- **TinyIPA** images are suitable for CI and testing environments and can be downloaded from <https://tarballs.openstack.org/ironic-python-agent/tinyipa/files/>.

Building from source

Check the `ironic-python-agent-builder` project for information on how to build `ironic-python-agent` ramdisks.

2.1.5 Integration with other OpenStack services

Configure the Identity service for the Bare Metal service

1. Create the Bare Metal service user (for example, `ironic`). The service uses this to authenticate with the Identity service. Use the `service` tenant and give the user the `admin` role:

```
$ openstack user create --password IRONIC_PASSWORD \  
  --email ironic@example.com ironic  
$ openstack role add --project service --user ironic admin
```

2. You must register the Bare Metal service with the Identity service so that other OpenStack services can locate it. To register the service:

```
$ openstack service create --name ironic --description \  
  "Ironic baremetal provisioning service" baremetal
```

3. Use the `id` property that is returned from the Identity service when registering the service (above), to create the endpoint, and replace `IRONIC_NODE` with your Bare Metal services API node:

```
$ openstack endpoint create --region RegionOne \
  baremetal admin http://$IRONIC_NODE:6385
$ openstack endpoint create --region RegionOne \
  baremetal public http://$IRONIC_NODE:6385
$ openstack endpoint create --region RegionOne \
  baremetal internal http://$IRONIC_NODE:6385
```

- You may delegate limited privileges related to the Bare Metal service to your Users by creating Roles with the OpenStack Identity service. By default, the Bare Metal service expects the `baremetal_admin` and `baremetal_observer` Roles to exist, in addition to the default `admin` Role. There is no negative consequence if you choose not to create these Roles. They can be created with the following commands:

```
$ openstack role create baremetal_admin
$ openstack role create baremetal_observer
```

If you choose to customize the names of Roles used with the Bare Metal service, do so by changing the `is_member`, `is_observer`, and `is_admin` policy settings in `/etc/ironic/policy.yaml`.

More complete documentation on managing Users and Roles within your OpenStack deployment are outside the scope of this document, but may be found [here](#).

- You can further restrict access to the Bare Metal service by creating a separate `baremetal` Project, so that Bare Metal resources (Nodes, Ports, etc) are only accessible to members of this Project:

```
$ openstack project create baremetal
```

At this point, you may grant read-only access to the Bare Metal service API without granting any other access by issuing the following commands:

```
$ openstack user create \
  --domain default --project-domain default --project baremetal \
  --password PASSWORD USERNAME
$ openstack role add \
  --user-domain default --project-domain default --project baremetal \
  --user USERNAME baremetal_observer
```

- Further documentation is available elsewhere for the `openstack` [command-line client](#) and the [Identity](#) service. A `policy.yaml.sample` file, which enumerates the services default policies, is provided for your convenience with the Bare Metal Service.

Configure the Compute service to use the Bare Metal service

The Compute service needs to be configured to use the Bare Metal services driver. The configuration file for the Compute service is typically located at `/etc/nova/nova.conf`.

Note: As of the Newton release, it is possible to have multiple `nova-compute` services running the ironic virtual driver (in `nova`) to provide redundancy. Bare metal nodes are mapped to the services via a hash ring. If a service goes down, the available bare metal nodes are remapped to different services.

Once active, a node will stay mapped to the same nova-compute even when it goes down. The node is unable to be managed through the Compute API until the service responsible returns to an active state.

The following configuration file must be modified on the Compute services controller nodes and compute nodes.

1. Change these configuration options in the Compute service configuration file (for example, `/etc/nova/nova.conf`):

```
[default]

# Defines which driver to use for controlling virtualization.
# Enable the ironic virt driver for this compute instance.
compute_driver=ironic.IronicDriver

# Amount of memory in MB to reserve for the host so that it is always
# available to host processes.
# It is impossible to reserve any memory on bare metal nodes, so set
# this to zero.
reserved_host_memory_mb=0

[filter_scheduler]

# Enables querying of individual hosts for instance information.
# Not possible for bare metal nodes, so set it to False.
track_instance_changes=False

[scheduler]

# This value controls how often (in seconds) the scheduler should
# attempt to discover new hosts that have been added to cells.
# If negative (the default), no automatic discovery will occur.
# As each bare metal node is represented by a separate host, it has
# to be discovered before the Compute service can deploy on it.
# The value here has to be carefully chosen based on a compromise
# between the enrollment speed and the load on the Compute scheduler.
# The recommended value of 2 minutes matches how often the Compute
# service polls the Bare Metal service for node information.
discover_hosts_in_cells_interval=120
```

Note: The alternative to setting the `discover_hosts_in_cells_interval` option is to run the following command on any Compute controller node after each node is enrolled:

```
nova-manage cell_v2 discover_hosts --by-service
```

2. Consider enabling the following option on controller nodes:

```
[filter_scheduler]
```

(continues on next page)

(continued from previous page)

```
# Enabling this option is beneficial as it reduces re-scheduling events
# for ironic nodes when scheduling is based on resource classes,
# especially for mixed hypervisor case with host_subset_size = 1.
# However enabling it will also make packing of VMs on hypervisors
# less dense even when scheduling weights are completely disabled.
#shuffle_best_same_weighed_hosts = false
```

3. Carefully consider the following option:

```
[compute]

# This option will cause nova-compute to set itself to a disabled state
# if a certain number of consecutive build failures occur. This will
# prevent the scheduler from continuing to send builds to a compute
# service that is consistently failing. In the case of bare metal
# provisioning, however, a compute service is rarely the cause of build
# failures. Furthermore, bare metal nodes, managed by a disabled
# compute service, will be remapped to a different one. That may cause
# the second compute service to also be disabled, and so on, until no
# compute services are active.
# If this is not the desired behavior, consider increasing this value or
# setting it to 0 to disable this behavior completely.
#consecutive_build_service_disable_threshold = 10
```

4. Change these configuration options in the `ironic` section. Replace:

- `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity Service
- `IRONIC_NODE` with the hostname or IP address of the `ironic-api` node
- `IDENTITY_IP` with the IP of the Identity server

```
[ironic]

# Ironic authentication type
auth_type=password

# Keystone API endpoint
auth_url=http://IDENTITY_IP:5000/v3

# Ironic keystone project name
project_name=service

# Ironic keystone admin name
username=ironic

# Ironic keystone admin password
password=IRONIC_PASSWORD

# Ironic keystone project domain
# or set project_domain_id
```

(continues on next page)

(continued from previous page)

```
project_domain_name=Default

# Ironic keystone user domain
# or set user_domain_id
user_domain_name=Default
```

5. On the Compute services controller nodes, restart the nova-scheduler process:

```
Fedora/RHEL8/CentOS8/SUSE:
sudo systemctl restart openstack-nova-scheduler

Ubuntu:
sudo service nova-scheduler restart
```

6. On the Compute services compute nodes, restart the nova-compute process:

```
Fedora/RHEL8/CentOS8/SUSE:
sudo systemctl restart openstack-nova-compute

Ubuntu:
sudo service nova-compute restart
```

Configure the Networking service for bare metal provisioning

You need to configure Networking so that the bare metal server can communicate with the Networking service for DHCP, PXE boot and other requirements. This section covers configuring Networking for a single flat network for bare metal provisioning.

It is recommended to use the baremetal ML2 mechanism driver and L2 agent for proper integration with the Networking service. Documentation regarding installation and configuration of the baremetal mechanism driver and L2 agent is available [here](#).

For use with [routed networks](#) the baremetal ML2 components are required.

Note: When the baremetal ML2 components are *not* used, ports in the Networking service will have status: DOWN, and binding_vif_type: binding_failed. This was always the status for Bare Metal service flat network interface ports prior to the introduction of the baremetal ML2 integration. For a non-routed network, bare metal servers can still be deployed and are functional, despite this port binding state in the Networking service.

You will also need to provide Bare Metal service with the MAC address(es) of each node that it is provisioning; Bare Metal service in turn will pass this information to Networking service for DHCP and PXE boot configuration. An example of this is shown in the [Enrollment](#) section.

1. Install the networking-baremetal ML2 mechanism driver and L2 agent in the Networking service.
2. Edit `/etc/neutron/plugins/ml2/ml2_conf.ini` and modify these:

```
[ml2]
type_drivers = flat
```

(continues on next page)

(continued from previous page)

```
tenant_network_types = flat
mechanism_drivers = openvswitch,baremetal

[ml2_type_flat]
flat_networks = physnet1

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.
↳OVSHybridIptablesFirewallDriver
enable_security_group = True

[ovs]
bridge_mappings = physnet1:br-eth2
# Replace eth2 with the interface on the neutron node which you
# are using to connect to the bare metal server
```

- Restart the `neutron-server` service, to load the new configuration.
- Create and edit `/etc/neutron/plugins/ml2/ironic_neutron_agent.ini` and add the required configuration. For example:

```
[ironic]
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = ironic
auth_url = http://identity-server.example.com/identity
auth_type = password
region_name = RegionOne
```

- Make sure the `ironic-neutron-agent` service is started.
- If `neutron-openvswitch-agent` runs with `ovs_neutron_plugin.ini` as the input config-file, edit `ovs_neutron_plugin.ini` to configure the bridge mappings by adding the `[ovs]` section described in the previous step, and restart the `neutron-openvswitch-agent`.
- Add the integration bridge to Open vSwitch:

```
$ ovs-vsctl add-br br-int
```

- Create the `br-eth2` network bridge to handle communication between the OpenStack services (and the Bare Metal services) and the bare metal nodes using `eth2`. Replace `eth2` with the interface on the network node which you are using to connect to the Bare Metal service:

```
$ ovs-vsctl add-br br-eth2
$ ovs-vsctl add-port br-eth2 eth2
```

- Restart the Open vSwitch agent:

```
# service neutron-plugin-openvswitch-agent restart
```

10. On restarting the Networking service Open vSwitch agent, the veth pair between the bridges br-int and br-eth2 is automatically created.

Your Open vSwitch bridges should look something like this after following the above steps:

```
$ ovs-vsctl show

Bridge br-int
  fail_mode: secure
  Port "int-br-eth2"
    Interface "int-br-eth2"
      type: patch
      options: {peer="phy-br-eth2"}
  Port br-int
    Interface br-int
      type: internal
Bridge "br-eth2"
  Port "phy-br-eth2"
    Interface "phy-br-eth2"
      type: patch
      options: {peer="int-br-eth2"}
  Port "eth2"
    Interface "eth2"
  Port "br-eth2"
    Interface "br-eth2"
      type: internal
ovs_version: "2.3.0"
```

11. Create the flat network on which you are going to launch the instances:

```
$ openstack network create --project $TENANT_ID sharednet1 --share \  
  --provider-network-type flat --provider-physical-network physnet1
```

12. Create the subnet on the newly created network:

```
$ openstack subnet create $SUBNET_NAME --network sharednet1 \  
  --subnet-range $NETWORK_CIDR --ip-version 4 --gateway $GATEWAY_IP \  
  --allocation-pool start=$START_IP,end=$END_IP --dhcp
```

Configuring services for bare metal provisioning using IPv6

Use of IPv6 addressing for baremetal provisioning requires additional configuration. This page covers the IPv6 specifics only. Please refer to *Configure tenant networks* and *Configure the Networking service for bare metal provisioning* for general networking configuration.

Configure ironic PXE driver for provisioning using IPv6 addressing

The ironic PXE driver operates in either IPv4 or IPv6 mode (IPv4 is the default). To enable IPv6 mode, set the `[pxe]/ip_version` option in the Bare Metal Services configuration file (`/etc/ironic/ironic.conf`) to 6.

Note: Support for dual mode IPv4 and IPv6 operations is planned for a future version of ironic.

Provisioning with IPv6 stateless addressing

When using stateless addressing DHCPv6 does not provide addresses to the client. DHCPv6 however provides other configuration via DHCPv6 options such as the `bootfile-url` and `bootfile-parameters`.

Once the PXE driver is set to operate in IPv6 mode no further configuration is required in the Baremetal Service.

Creating networks and subnets in the Networking Service

When creating the Baremetal Service network(s) and subnet(s) in the Networking Services, subnets should have `ipv6-address-mode` set to `dhcpv6-stateless` and `ip-version` set to 6. Depending on whether a router in the Networking Service is providing RAs (Router Advertisements) or not, the `ipv6-ra-mode` for the subnet(s) should either be set to `dhcpv6-stateless` or be left unset.

Note: If `ipv6-ra-mode` is left unset, an external router on the network is expected to provide RAs with the appropriate flags set for automatic addressing and other configuration.

Provisioning with IPv6 stateful addressing

When using stateful addressing DHCPv6 is providing both addresses and other configuration via DHCPv6 options such as the `bootfile-url` and `bootfile-parameters`.

The identity-association (IA) construct used by DHCPv6 is challenging when booting over the network. Firmware, and ramdisks typically end up using different DUID/IAID combinations and it is not always possible for one chain- booting stage to release its address before giving control to the next step. In case the DHCPv6 server is configured with static reservations only the result is that booting will fail because the DHCPv6 server has no addresses available. To get past this issue either configure the DHCPv6 server with multiple address reservations for each host, or use a dynamic range.

Note: Support for multiple address reservations requires dnsmasq version 2.81 or later. Some distributions may backport this feature to earlier dnsmasq version as part of the packaging, check the distributions release notes.

If a different (not dnsmasq) DHCPv6 server backend is used with the Networking service, use of multiple address reservations might not work.

Using the flat network interface

Due to the identity-association challenges with DHCPv6 provisioning using the flat network interface is not recommended. When ironic operates with the flat network interface the server instance port is used for provisioning and other operations. Ironic will not use multiple address reservations in this scenario. Because of this **it will not work in most cases**.

Using the neutron network interface

When using the neutron network interface the Baremetal Service will allocate multiple IPv6 addresses (4 addresses per port by default) on the service networks used for provisioning, cleaning, rescue and introspection. The number of addresses allocated can be controlled via the [neutron]/dhcpv6_stateful_address_count option in the Bare Metal Services configuration file (/etc/ironic/ironic.conf). Using multiple address reservations ensures that the DHCPv6 server can lease addresses to each step.

To enable IPv6 provisioning on neutron flat provider networks with no switch management, the local_link_connection field of baremetal ports must be set to {'network_type': 'unmanaged'}. The following example shows how to set the local_link_connection for operation on unmanaged networks:

```
baremetal port set \  
  --local-link-connection network_type=unmanaged <port-uuid>
```

The use of multiple IPv6 addresses must also be enabled in the Networking Services dhcp agent configuration (/etc/neutron/dhcp_agent.ini) by setting the option [DEFAULT]/dnsmasq_enable_addr6_list to True (default False in Ussuri release).

Note: Support for multiple IPv6 address reservations in the dnsmasq backend was added to the Networking Service Ussuri release. It was also backported to the stable Train release.

Creating networks and subnets in the Networking Service

When creating the ironic service network(s) and subnet(s) in the Networking Service, subnets should have `ipv6-address-mode` set to `dhcpv6-stateful` and `ip-version` set to `6`. Depending on whether a router in the Networking Service is providing RAs (Router Advertisements) or not, the `ipv6-ra-mode` for the subnet(s) should be set to either `dhcpv6-stateful` or be left unset.

Note: If `ipv6-ra-mode` is left unset, an external router on the network is expected to provide RAs with the appropriate flags set for managed addressing and other configuration.

Configure the Image service for temporary URLs

Some drivers of the Baremetal service (in particular, any drivers using *Direct deploy* or *Ansible deploy* interfaces, and some virtual media drivers) require target user images to be available over clean HTTP(S) URL with no authentication involved (neither username/password-based, nor token-based).

When using the Baremetal service integrated in OpenStack, this can be achieved by specific configuration of the Image service and Object Storage service as described below.

1. Configure the Image service to have object storage as a backend for storing images. For more details, please refer to the Image service configuration guide.

Note: When using Ceph+RadosGW for Object Storage service, images stored in Image service must be available over Object Storage service as well.

2. Enable TempURLs for the Object Storage account used by the Image service for storing images in the Object Storage service.
 1. Check if TempURLs are enabled:

```
# executed under credentials of the user used by Image service
# to access Object Storage service
$ openstack object store account show
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| Account    | AUTH_bc39f1d9dcf9486899088007789ae643 |
| Bytes      | 536661727                                 |
| Containers | 1                                         |
| Objects    | 19                                        |
| properties | Temp-Url-Key='secret'                    |
+-----+-----+
```

2. If property `Temp-Url-Key` is set, note its value.
3. If property `Temp-Url-Key` is not set, you have to configure it (`secret` is used in the example below for the value):

```
$ openstack object store account set --property Temp-Url-Key=secret
```

3. Optionally, configure the ironic-conductor service. The default configuration assumes that:
 1. the Object Storage service is implemented by `swift`,
 2. the Object Storage service URL is available from the service catalog,
 3. the project, used by the Image service to access the Object Storage, is the same as the project, used by the Bare Metal service to access it,
 4. the container, used by the Image service, is called `glance`.

If any of these assumptions do not hold, you may want to change your configuration file (typically located at `/etc/ironic/ironic.conf`), for example:

```
[glance]
swift_endpoint_url = http://openstack/swift
swift_account = AUTH_bc39f1d9dcf9486899088007789ae643
swift_container = glance
swift_temp_url_key = secret
```

4. (Re)start the ironic-conductor service.

Enabling HTTPS

Enabling HTTPS in Swift

The drivers using virtual media use swift for storing boot images and node configuration information (contains sensitive information for Ironic conductor to provision bare metal hardware). By default, HTTPS is not enabled in swift. HTTPS is required to encrypt all communication between swift and Ironic conductor and swift and bare metal (via virtual media). It can be enabled in one of the following ways:

- Using an SSL termination proxy
- Using native SSL support in swift (recommended only for testing purpose by swift).

Enabling HTTPS in Image service

Ironic drivers usually use Image service during node provisioning. By default, image service does not use HTTPS, but it is required for secure communication. It can be enabled by making the following changes to `/etc/glance/glance-api.conf`:

1. Configuring SSL support
2. Restart the glance-api service:

```
Fedora/RHEL8/CentOS8/SUSE:
    sudo systemctl restart openstack-glance-api

Debian/Ubuntu:
    sudo service glance-api restart
```

See the [Glance documentation](#), for more details on the Image service.

Enabling HTTPS communication between Image service and Object storage

This section describes the steps needed to enable secure HTTPS communication between Image service and Object storage when Object storage is used as the Backend.

To enable secure HTTPS communication between Image service and Object storage follow these steps:

1. *Enabling HTTPS in Swift*
2. *Configure Swift Storage Backend*
3. *Enabling HTTPS in Image service*

Enabling HTTPS communication between Image service and Bare Metal service

This section describes the steps needed to enable secure HTTPS communication between Image service and Bare Metal service.

To enable secure HTTPS communication between Bare Metal service and Image service follow these steps:

1. Edit `/etc/ironic/ironic.conf`:

```
[glance]
...
glance_cafile=/path/to/certfile
```

Note: `glance_cafile` is an optional path to a CA certificate bundle to be used to validate the SSL certificate served by Image service.

2. If not using the keystone service catalog for the Image service API endpoint discovery, also edit the `endpoint_override` option to point to HTTPS URL of image service (replace `<GLANCE_API_ADDRESS>` with `hostname[:port][path]` of the Image service endpoint):

```
[glance]
...
endpoint_override = https://<GLANCE_API_ADDRESS>
```

3. Restart `ironic-conductor` service:

```
Fedora/RHEL8/CentOS8/SUSE:
    sudo systemctl restart openstack-ironic-conductor

Debian/Ubuntu:
    sudo service ironic-conductor restart
```

Configure the Bare Metal service for cleaning

Note: If you configured the Bare Metal service to do *Automated cleaning* (which is enabled by default), you will need to set the `cleaning_network` configuration option.

1. Note the network UUID (the `id` field) of the network you created in *Configure the Networking service for bare metal provisioning* or another network you created for cleaning:

```
$ openstack network list
```

2. Configure the cleaning network UUID via the `cleaning_network` option in the Bare Metal service configuration file (`/etc/ironic/ironic.conf`). In the following, replace `NETWORK_UUID` with the UUID you noted in the previous step:

```
[neutron]
cleaning_network = NETWORK_UUID
```

3. Restart the Bare Metal services `ironic-conductor`:

```
Fedora/RHEL8/CentOS8/SUSE:
sudo systemctl restart openstack-ironic-conductor

Ubuntu:
sudo service ironic-conductor restart
```

Configure tenant networks

Below is an example flow of how to set up the Bare Metal service so that node provisioning will happen in a multi-tenant environment (which means using the `neutron` network interface as stated above):

1. Network interfaces can be enabled on `ironic-conductor` by adding them to the `enabled_network_interfaces` configuration option under the `default` section of the configuration file:

```
[DEFAULT]
...
enabled_network_interfaces=noop,flat,neutron
```

Keep in mind that, ideally, all `ironic-conductors` should have the same list of enabled network interfaces, but it may not be the case during `ironic-conductor` upgrades. This may cause problems if one of the `ironic-conductors` dies and some node that is taken over is mapped to an `ironic-conductor` that does not support the nodes network interface. Any actions that involve calling the nodes driver will fail until that network interface is installed and enabled on that `ironic-conductor`.

2. It is recommended to set the default network interface via the `default_network_interface` configuration option under the `default` section of the configuration file:

```
[DEFAULT]
...
default_network_interface=neutron
```

This default value will be used for all nodes that don't have a network interface explicitly specified in the creation request.

If this configuration option is not set, the default network interface is determined by looking at the `[dhcp]dhcp_provider` configuration option value. If it is `neutron`, then `flat` network interface becomes the default, otherwise `noop` is the default.

3. Define a provider network in the Networking service, which we shall refer to as the provisioning network. Using the `neutron` network interface requires that `provisioning_network` and `cleaning_network` configuration options are set to valid identifiers (UUID or name) of networks in the Networking service. If these options are not set correctly, cleaning or provisioning will fail to start. There are two ways to set these values:

- Under the `neutron` section of ironic configuration file:

```
[neutron]
cleaning_network = $CLEAN_UUID_OR_NAME
provisioning_network = $PROVISION_UUID_OR_NAME
```

- Under `provisioning_network` and `cleaning_network` keys of the nodes `driver_info` field as `driver_info['provisioning_network']` and `driver_info['cleaning_network']` respectively.

Note: If these `provisioning_network` and `cleaning_network` values are not specified in nodes `driver_info` then ironic falls back to the configuration in the `neutron` section.

Please refer to *Configure the Bare Metal service for cleaning* for more information about cleaning.

Warning: Please make sure that the Bare Metal service has exclusive access to the provisioning and cleaning networks. Spawning instances by non-admin users in these networks and getting access to the Bare Metal services control plane is a security risk. For this reason, the provisioning and cleaning networks should be configured as non-shared networks in the `admin` tenant.

Note: When using the `flat` network interface, bare metal instances are normally spawned onto the provisioning network. This is not supported with the `neutron` interface and the deployment will fail. Please ensure a different network is chosen in the Networking service when a bare metal instance is booted from the Compute service.

Note: The provisioning and cleaning networks may be the same network or distinct networks. To ensure that communication between the Bare Metal service and the deploy ramdisk works, it is important to ensure that security groups are disabled for these networks, *or* that the default security groups allow:

- DHCP
- TFTP
- egress port used for the Bare Metal service (6385 by default)

- ingress port used for ironic-python-agent (9999 by default)
 - if using *Direct deploy*, the egress port used for the Object Storage service or the local HTTP server (typically 80 or 443)
 - if using iPXE, the egress port used for the HTTP server running on the ironic-conductor nodes (typically 80).
-

4. This step is optional and applicable only if you want to use security groups during provisioning and/or cleaning of the nodes. If not specified, default security groups are used.

1. Define security groups in the Networking service, to be used for provisioning and/or cleaning networks.
2. Add the list of these security group UUIDs under the `neutron` section of ironic-conductors configuration file as shown below:

```
[neutron]
...
cleaning_network=$CLEAN_UUID_OR_NAME
cleaning_network_security_groups=[$LIST_OF_CLEAN_SECURITY_GROUPS]
provisioning_network=$PROVISION_UUID_OR_NAME
provisioning_network_security_groups=[$LIST_OF_PROVISION_SECURITY_
↪GROUPS]
```

Multiple security groups may be applied to a given network, hence, they are specified as a list. The same security group(s) could be used for both provisioning and cleaning networks.

Warning: If security groups are configured as described above, do not set the `port_security_enabled` flag to `False` for the corresponding Networking services network or port. This will cause the deploy to fail.

For example: if `provisioning_network_security_groups` configuration option is used, ensure that `port_security_enabled` flag for the provisioning network is set to `True`. This flag is set to `True` by default; make sure not to override it by manually setting it to `False`.

5. Install and configure a compatible ML2 mechanism driver which supports bare metal provisioning for your switch. See [ML2 plugin configuration manual](#) for details.
6. Restart the ironic-conductor and ironic-api services after the modifications:

- Fedora/RHEL8/CentOS8:

```
sudo systemctl restart openstack-ironic-api
sudo systemctl restart openstack-ironic-conductor
```

- Ubuntu:

```
sudo service ironic-api restart
sudo service ironic-conductor restart
```

7. Make sure that the ironic-conductor is reachable over the provisioning network by trying to download a file from a TFTP server on it, from some non-control-plane server in that network:

```
tftp $TFTP_IP -c get $FILENAME
```

where FILENAME is the file located at the TFTP server.

See *Multi-tenancy in the Bare Metal service* for required node configuration.

Add images to the Image service

1. Build or download the user images as described in *Create user images for the Bare Metal service*.
2. Add the user images to the Image service

Load all the images created in the below steps into the Image service, and note the image UUIDs in the Image service for each one as it is generated.

For *partition images*:

- Add the kernel and ramdisk images to the Image service:

```
$ openstack image create my-kernel --public \
  --disk-format aki --container-format aki --file my-image.vmlinuz
```

Store the image uuid obtained from the above step as MY_VMLINUZ_UUID.

```
$ openstack image create my-image.initrd --public \
  --disk-format ari --container-format ari --file my-image.initrd
```

Store the image UUID obtained from the above step as MY_INITRD_UUID.

- Add the *my-image* to the Image service which is going to be the OS that the user is going to run. Also associate the above created images with this OS image. These two operations can be done by executing the following command:

```
$ openstack image create my-image --public \
  --disk-format qcow2 --container-format bare --property \
  kernel_id=$MY_VMLINUZ_UUID --property \
  ramdisk_id=$MY_INITRD_UUID --file my-image.qcow2
```

For *whole disk images*, skip uploading and configuring kernel and ramdisk images completely, proceed directly to uploading the main image:

```
$ openstack image create my-whole-disk-image --public \
  --disk-format qcow2 --container-format bare \
  --file my-whole-disk-image.qcow2
```

Warning: The kernel/initramfs pair must not be set for whole disk images, otherwise they'll be mistaken for partition images.

3. Build or download the deploy images

The deploy images are used initially for preparing the server (creating disk partitions) before the actual OS can be deployed.

There are several methods to build or download deploy images, please read the *Building or downloading a deploy ramdisk image* section.

4. Add the deploy images to the Image service

Add the deployment kernel and ramdisk images to the Image service:

```
$ openstack image create deploy-vmlinux --public \  
--disk-format aki --container-format aki \  
--file ironic-python-agent.vmlinux
```

Store the image UUID obtained from the above step as `DEPLOY_VMLINUX_UUID`.

```
$ openstack image create deploy-initrd --public \  
--disk-format ari --container-format ari \  
--file ironic-python-agent.initramfs
```

Store the image UUID obtained from the above step as `DEPLOY_INITRD_UUID`.

Create flavors for use with the Bare Metal service

You'll need to create a special bare metal flavor in the Compute service. The flavor is mapped to the bare metal node through the nodes `resource_class` field (available starting with Bare Metal API version 1.21). A flavor can request *exactly one* instance of a bare metal resource class.

Note that when creating the flavor, it's useful to add the `RAM_MB` and `CPU` properties as a convenience to users, although they are not used for scheduling. The `DISK_GB` property is also not used for scheduling, but is still used to determine the root partition size.

1. Change these to match your hardware:

```
$ RAM_MB=1024  
$ CPU=2  
$ DISK_GB=100
```

2. Create the bare metal flavor by executing the following command:

```
$ openstack flavor create --ram $RAM_MB --vcpus $CPU --disk $DISK_GB \  
my-baremetal-flavor
```

Note: You can add `--id <id>` to specify an ID for the flavor.

See the [docs on this command](#) for other options that may be specified.

After creation, associate each flavor with one custom resource class. The name of a custom resource class that corresponds to a nodes resource class (in the Bare Metal service) is:

- the bare metal nodes resource class all upper-cased
- prefixed with `CUSTOM_`
- all punctuation replaced with an underscore

For example, if the resource class is named `baremetal-small`, associate the flavor with this custom resource class via:

```
$ openstack flavor set --property resources:CUSTOM_BAREMETAL_SMALL=1 my-
↪baremetal-flavor
```

Another set of flavor properties must be used to disable scheduling based on standard properties for a bare metal flavor:

```
$ openstack flavor set --property resources:VCPU=0 my-baremetal-flavor
$ openstack flavor set --property resources:MEMORY_MB=0 my-baremetal-flavor
$ openstack flavor set --property resources:DISK_GB=0 my-baremetal-flavor
```

Example

If you want to define a class of nodes called `baremetal.with-GPU`, start with tagging some nodes with it:

```
$ baremetal node set <node> --resource-class baremetal.with-GPU
```

Warning: It is possible to **add** a resource class to active nodes, but it is not possible to **replace** an existing resource class on them.

Then you can update your flavor to request the resource class instead of the standard properties:

```
$ openstack flavor set --property resources:CUSTOM_BAREMETAL_WITH_GPU=1 my-
↪baremetal-flavor
$ openstack flavor set --property resources:VCPU=0 my-baremetal-flavor
$ openstack flavor set --property resources:MEMORY_MB=0 my-baremetal-flavor
$ openstack flavor set --property resources:DISK_GB=0 my-baremetal-flavor
```

Note how `baremetal.with-GPU` in the nodes `resource_class` field becomes `CUSTOM_BAREMETAL_WITH_GPU` in the flavors properties.

Scheduling based on traits

Starting with the Queens release, the Compute service supports scheduling based on qualitative attributes using traits. Starting with Bare Metal REST API version 1.37, it is possible to assign a list of traits to each bare metal node. Traits assigned to a bare metal node will be assigned to the corresponding resource provider in the Compute service placement API.

When creating a flavor in the Compute service, required traits may be specified via flavor properties. The Compute service will then schedule instances only to bare metal nodes with all of the required traits.

Traits can be either standard or custom. Standard traits are listed in the [os_traits library](#). Custom traits must meet the following requirements:

- prefixed with `CUSTOM_`
- contain only upper case characters A to Z, digits 0 to 9, or underscores
- no longer than 255 characters in length

A bare metal node can have a maximum of 50 traits.

Example

To add the standard trait `HW_CPU_X86_VMX` and a custom trait `CUSTOM_TRAIT1` to a node:

```
$ baremetal node add trait <node> CUSTOM_TRAIT1 HW_CPU_X86_VMX
```

Then, update the flavor to require these traits:

```
$ openstack flavor set --property trait:CUSTOM_TRAIT1=required my-baremetal-  
↪flavor  
$ openstack flavor set --property trait:HW_CPU_X86_VMX=required my-baremetal-  
↪flavor
```

2.1.6 Set up the drivers for the Bare Metal service

Enabling drivers and hardware types

Introduction

The Bare Metal service delegates actual hardware management to **drivers**. *Drivers*, also called *hardware types*, consist of *hardware interfaces*: sets of functionality dealing with some aspect of bare metal provisioning in a vendor-specific way. There are generic **hardware types** (eg. `redfish` and `ipmi`), and vendor-specific ones (eg. `ilo` and `irmc`).

Note: Starting with the Rocky release, the terminologies *driver*, *dynamic driver*, and *hardware type* have the same meaning in the scope of Bare Metal service.

Enabling hardware types

Hardware types are enabled in the configuration file of the **ironic-conductor** service by setting the `enabled_hardware_types` configuration option, for example:

```
[DEFAULT]  
enabled_hardware_types = ipmi,redfish
```

Due to the drivers dynamic nature, they also require configuring enabled hardware interfaces.

Note: All available hardware types and interfaces are listed in `setup.cfg` file in the source code tree.

Enabling hardware interfaces

There are several types of hardware interfaces:

bios manages configuration of the BIOS settings of a bare metal node. This interface is vendor-specific and can be enabled via the `enabled_bios_interfaces` option:

```
[DEFAULT]
enabled_hardware_types = <hardware_type_name>
enabled_bios_interfaces = <bios_interface_name>
```

See *BIOS Configuration* for details.

boot manages booting of both the deploy ramdisk and the user instances on the bare metal node. See *Boot interfaces* for details.

Boot interface implementations are often vendor specific, and can be enabled via the `enabled_boot_interfaces` option:

```
[DEFAULT]
enabled_hardware_types = ipmi,ilo
enabled_boot_interfaces = pxe,ilo-virtual-media
```

Boot interfaces with `pxe` in their name require *Configuring PXE and iPXE*. There are also a few hardware-specific boot interfaces - see *Drivers, Hardware Types and Hardware Interfaces* for their required configuration.

console manages access to the serial console of a bare metal node. See *Configuring Web or Serial Console* for details.

deploy defines how the image gets transferred to the target disk. See *Deploy Interfaces* for an explanation of the difference between supported deploy interfaces.

The deploy interfaces can be enabled as follows:

```
[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_deploy_interfaces = direct,ramdisk
```

Note: The `direct` deploy interface requires the Object Storage service or an HTTP service

inspect implements fetching hardware information from nodes. Can be implemented out-of-band (via contacting the nodes BMC) or in-band (via booting a ramdisk on a node). The latter implementation is called `inspector` and uses a separate service called `ironic-inspector`. Example:

```
[DEFAULT]
enabled_hardware_types = ipmi,ilo,irmc
enabled_inspect_interfaces = ilo,irmc,inspector
```

See *Hardware Inspection* for more details.

management provides additional hardware management actions, like getting or setting boot devices. This interface is usually vendor-specific, and its name often matches the name of the hardware type (with `ipmi tool` being a notable exception). For example:

[DEFAULT]

```
enabled_hardware_types = ipmi,redfish,ilo,irmc
enabled_management_interfaces = ipmitool,redfish,ilo,irmc
```

Using `ipmitool` requires *Configuring IPMI support*. See *Drivers, Hardware Types and Hardware Interfaces* for the required configuration of each driver.

network connects/disconnects bare metal nodes to/from virtual networks. See *Configure tenant networks* for more details.

power runs power actions on nodes. Similar to the management interface, it is usually vendor-specific, and its name often matches the name of the hardware type (with `ipmitool` being again an exception). For example:

[DEFAULT]

```
enabled_hardware_types = ipmi,redfish,ilo,irmc
enabled_power_interfaces = ipmitool,redfish,ilo,irmc
```

Using `ipmitool` requires *Configuring IPMI support*. See *Drivers, Hardware Types and Hardware Interfaces* for the required configuration of each driver.

raid manages building and tearing down RAID on nodes. Similar to inspection, it can be implemented either out-of-band or in-band (via agent implementation). See *RAID Configuration* for details. For example:

[DEFAULT]

```
enabled_hardware_types = ipmi,redfish,ilo,irmc
enabled_raid_interfaces = agent,no-raid
```

storage manages the interaction with a remote storage subsystem, such as the Block Storage service, and helps facilitate booting from a remote volume. This interface ensures that volume target and connector information is updated during the lifetime of a deployed instance. See *Boot From Volume* for more details.

This interface defaults to a `noop` driver as it is considered an opt-in interface which requires additional configuration by the operator to be usable.

For example:

[DEFAULT]

```
enabled_hardware_types = ipmi,irmc
enabled_storage_interfaces = cinder,noop
```

vendor is a place for vendor extensions to be exposed in API. See *Vendor Methods* for details.

[DEFAULT]

```
enabled_hardware_types = ipmi,redfish,ilo,irmc
enabled_vendor_interfaces = ipmitool,no-vendor
```

Here is a complete configuration example, enabling two generic protocols, IPMI and Redfish, with a few additional features:

[DEFAULT]

```
enabled_hardware_types = ipmi,redfish
```

(continues on next page)

(continued from previous page)

```

enabled_boot_interfaces = pxe
enabled_console_interfaces = ipmitool-socat,no-console
enabled_deploy_interfaces = direct
enabled_inspect_interfaces = inspector
enabled_management_interfaces = ipmitool,redfish
enabled_network_interfaces = flat,neutron
enabled_power_interfaces = ipmitool,redfish
enabled_raid_interfaces = agent
enabled_storage_interfaces = cinder,noop
enabled_vendor_interfaces = ipmitool,no-vendor

```

Note that some interfaces have implementations named `no-<TYPE>` where `<TYPE>` is the interface type. These implementations do nothing and return errors when used from API.

Hardware interfaces in multi-conductor environments

When enabling hardware types and their interfaces, make sure that for every enabled hardware type, the whole set of enabled interfaces matches for all conductors. However, different conductors can have different hardware types enabled.

For example, you can have two conductors with the following configuration respectively:

```

[DEFAULT]
enabled_hardware_types = ipmi
enabled_deploy_interfaces = direct
enabled_power_interfaces = ipmitool
enabled_management_interfaces = ipmitool

```

```

[DEFAULT]
enabled_hardware_types = redfish
enabled_deploy_interfaces = ansible
enabled_power_interfaces = redfish
enabled_management_interfaces = redfish

```

But you cannot have two conductors with the following configuration respectively:

```

[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_deploy_interfaces = direct
enabled_power_interfaces = ipmitool,redfish
enabled_management_interfaces = ipmitool,redfish

```

```

[DEFAULT]
enabled_hardware_types = redfish
enabled_deploy_interfaces = ansible
enabled_power_interfaces = redfish
enabled_management_interfaces = redfish

```

This is because the `redfish` hardware type will have different enabled `deploy` interfaces on these conductors. It would have been fine, if the second conductor had `enabled_deploy_interfaces = direct`

instead of `ansible`.

This situation is not detected by the Bare Metal service, but it can cause inconsistent behavior in the API, when node functionality will depend on which conductor it gets assigned to.

Note: We don't treat this as an error, because such *temporary* inconsistency is inevitable during a rolling upgrade or a configuration update.

Configuring interface defaults

When an operator does not provide an explicit value for one of the interfaces (when creating a node or updating its driver), the default value is calculated as described in *Defaults for hardware interfaces*. It is also possible to override the defaults for any interfaces by setting one of the options named `default_<IFACE>_interface`, where `<IFACE>` is the interface name. For example:

```
[DEFAULT]
default_deploy_interface = direct
default_network_interface = neutron
```

This configuration forces the default `deploy` interface to be `direct` and the default `network` interface to be `neutron` for all hardware types.

The defaults are calculated and set on a node when creating it or updating its hardware type. Thus, changing these configuration options has no effect on existing nodes.

Warning: The default interface implementation must be configured the same way across all conductors in the cloud, except maybe for a short period of time during an upgrade or configuration update. Otherwise the default implementation will depend on which conductor handles which node, and this mapping is not predictable or even persistent.

Warning: These options should be used with care. If a hardware type does not support the provided default implementation, its users will have to always provide an explicit value for this interface when creating a node.

Configuring PXE and iPXE

DHCP server setup

A DHCP server is required by PXE/iPXE client. You need to follow steps below.

1. Set the `[dhcp]/dhcp_provider` to `neutron` in the Bare Metal Services configuration file (`/etc/ironic/ironic.conf`):

Note: Refer *Configure tenant networks* for details. The `dhcp_provider` configuration is already set by the configuration defaults, and when you create subnet, DHCP is also enabled if you do not

add any dhcp options at openstack subnet create command.

2. Enable DHCP in the subnet of PXE network.
 3. Set the ip address range in the subnet for DHCP.
-

Note: Refer *Configure the Networking service for bare metal provisioning* for details about the two precedent steps.

4. Connect the openstack DHCP agent to the external network through the OVS bridges and the interface eth2.
-

Note: Refer *Configure the Networking service for bare metal provisioning* for details. You do not require this part if br-int, br-eth2 and eth2 are already connected.

5. Configure the host ip at br-eth2. If it locates at eth2, do below:

```
ip addr del 192.168.2.10/24 dev eth2
ip addr add 192.168.2.10/24 dev br-eth2
```

Note: Replace eth2 with the interface on the network node which you are using to connect to the Bare Metal service.

TFTP server setup

In order to deploy instances via PXE, a TFTP server needs to be set up on the Bare Metal service nodes which run the `ironic-conductor`.

1. Make sure the tftp root directory exist and can be written to by the user the `ironic-conductor` is running as. For example:

```
sudo mkdir -p /tftpboot
sudo chown -R ironic /tftpboot
```

2. Install tftp server:

Ubuntu:

```
sudo apt-get install xinetd tftpd-hpa
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install tftp-server xinetd
```

SUSE:

```
sudo zypper install tftp xinetd
```

- Using xinetd to provide a tftp server setup to serve /tftpboot. Create or edit /etc/xinetd.d/tftp as below:

```
service tftp
{
    protocol          = udp
    port              = 69
    socket_type       = dgram
    wait              = yes
    user              = root
    server             = /usr/sbin/in.tftpd
    server_args        = -v -v -v -v -v --map-file /tftpboot/map-file /tftpboot
    disable            = no
    # This is a workaround for Fedora, where TFTP will listen only on
    # IPv6 endpoint, if IPv4 flag is not used.
    flags              = IPv4
}
```

and restart the xinetd service:

Ubuntu:

```
sudo service xinetd restart
```

Fedora/RHEL8/CentOS8/SUSE:

```
sudo systemctl restart xinetd
```

Note: In certain environments the networks MTU may cause TFTP UDP packets to get fragmented. Certain PXE firmwares struggle to reconstruct the fragmented packets which can cause significant slow down or even prevent the server from PXE booting. In order to avoid this, TFTPd provides an option to limit the packet size so that they do not get fragmented. To set this additional option in the server_args above:

```
--blocksize <MAX MTU minus 32>
```

- Create a map file in the tftp boot directory (/tftpboot):

```
echo 're ^(/tftpboot/) /tftpboot/\2' > /tftpboot/map-file
echo 're ^/tftpboot/ /tftpboot/' >> /tftpboot/map-file
echo 're ^(^/) /tftpboot/\1' >> /tftpboot/map-file
echo 're ^([^\]) /tftpboot/\1' >> /tftpboot/map-file
```

UEFI PXE - Grub setup

In order to deploy instances with PXE on bare metal nodes which support UEFI, perform these additional steps on the ironic conductor node to configure the PXE UEFI environment.

1. Install Grub2 and shim packages:

Ubuntu (16.04LTS and later):

```
sudo apt-get install grub-efi-amd64-signed shim-signed
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install grub2-efi shim
```

SUSE:

```
sudo zypper install grub2-x86_64-efi shim
```

2. Copy grub and shim boot loader images to /tftpboot directory:

Ubuntu (16.04LTS and later):

```
sudo cp /usr/lib/shim/shim.efi.signed /tftpboot/bootx64.efi
sudo cp /usr/lib/grub/x86_64-efi-signed/grubnetx64.efi.signed /tftpboot/
↪grubx64.efi
```

Fedora:

```
sudo cp /boot/efi/EFI/fedora/shim.efi /tftpboot/bootx64.efi
sudo cp /boot/efi/EFI/fedora/grubx64.efi /tftpboot/grubx64.efi
```

RHEL8/CentOS8:

```
sudo cp /boot/efi/EFI/centos/shim.efi /tftpboot/bootx64.efi
sudo cp /boot/efi/EFI/centos/grubx64.efi /tftpboot/grubx64.efi
```

SUSE:

```
sudo cp /usr/lib64/efi/shim.efi /tftpboot/bootx64.efi
sudo cp /usr/lib/grub2/x86_64-efi/grub.efi /tftpboot/grubx64.efi
```

3. Create master grub.cfg:

Ubuntu: Create grub.cfg under /tftpboot/grub directory:

```
GRUB_DIR=/tftpboot/grub
```

Fedora: Create grub.cfg under /tftpboot/EFI/fedora directory:

```
GRUB_DIR=/tftpboot/EFI/fedora
```

RHEL8/CentOS8: Create grub.cfg under /tftpboot/EFI/centos directory:

```
GRUB_DIR=/tftpboot/EFI/centos
```

SUSE: Create grub.cfg under /tftpboot/boot/grub directory:

```
GRUB_DIR=/tftpboot/boot/grub
```

Create directory GRUB_DIR:

```
sudo mkdir -p $GRUB_DIR
```

This file is used to redirect grub to baremetal node specific config file. It redirects it to specific grub config file based on DHCP IP assigned to baremetal node.

```
set default=master
set timeout=5
set hidden_timeout_quiet=false

menuentry "master" {
configfile /tftpboot/$net_default_mac.conf
}
```

Change the permission of grub.cfg:

```
sudo chmod 644 $GRUB_DIR/grub.cfg
```

4. Update the bare metal node with `boot_mode:uefi` capability in nodes properties field. See [Boot mode support](#) for details.
5. Make sure that bare metal node is configured to boot in UEFI boot mode and boot device is set to `network/pxe`.

Note: Some drivers, e.g. `ilo`, `irmc` and `redfish`, support automatic setting of the boot mode during deployment. This step is not required for them. Please check [Drivers](#), [Hardware Types and Hardware Interfaces](#) for information on whether your driver requires manual UEFI configuration.

Legacy BIOS - Syslinux setup

In order to deploy instances with PXE on bare metal using Legacy BIOS boot mode, perform these additional steps on the ironic conductor node.

1. Install the syslinux package with the PXE boot images:

Ubuntu (16.04LTS and later):

```
sudo apt-get install syslinux-common pxelinux
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install syslinux-tftpboot
```

SUSE:


```
sudo zypper install syslinux
```

- Copy the PXE image to `/tftpboot`. The PXE image might be found at¹:

Ubuntu (16.04LTS and later):

```
sudo cp /usr/lib/PXELINUX/pxelinux.0 /tftpboot
```

RHEL8/CentOS8/SUSE:

```
sudo cp /usr/share/syslinux/pxelinux.0 /tftpboot
```

- If whole disk images need to be deployed via PXE-netboot, copy the `chain.c32` image to `/tftpboot` to support it:

Ubuntu (16.04LTS and later):

```
sudo cp /usr/lib/syslinux/modules/bios/chain.c32 /tftpboot
```

Fedora:

```
sudo cp /boot/extlinux/chain.c32 /tftpboot
```

RHEL8/CentOS8/SUSE:

```
sudo cp /usr/share/syslinux/chain.c32 /tftpboot/
```

- If the version of `syslinux` is **greater than 4** we also need to make sure that we copy the library modules into the `/tftpboot` directory^{2?}. For example, for Ubuntu run:

```
sudo cp /usr/lib/syslinux/modules/*/ldlinux.* /tftpboot
```

- Update the bare metal node with `boot_mode: bios` capability in nodes properties field. See *Boot mode support* for details.
- Make sure that bare metal node is configured to boot in Legacy BIOS boot mode and boot device is set to `network/pxe`.

iPXE setup

If you will be using iPXE to boot instead of PXE, iPXE needs to be set up on the Bare Metal service node(s) where `ironic-conductor` is running.

- Make sure these directories exist and can be written to by the user the `ironic-conductor` is running as. For example:

```
sudo mkdir -p /tftpboot
sudo mkdir -p /httpboot
```

(continues on next page)

¹ On **Fedora/RHEL** the `syslinux-tftpboot` package already installs the library modules and PXE image at `/tftpboot`. If the TFTP server is configured to listen to a different directory you should copy the contents of `/tftpboot` to the configured directory

² http://www.syslinux.org/wiki/index.php/Library_modules

(continued from previous page)

```
sudo chown -R ironic /tftpboot
sudo chown -R ironic /httpboot
```

2. Create a map file in the tftp boot directory (/tftpboot):

```
echo 'r ^([^\s/]) /tftpboot/\1' > /tftpboot/map-file
echo 'r ^(/tftpboot/) /tftpboot/\2' >> /tftpboot/map-file
```

3. Set up TFTP and HTTP servers.

These servers should be running and configured to use the local /tftpboot and /httpboot directories respectively, as their root directories. (Setting up these servers is outside the scope of this install guide.)

These root directories need to be mounted locally to the `ironic-conductor` services, so that the services can access them.

The Bare Metal services configuration file (`/etc/ironic/ironic.conf`) should be edited accordingly to specify the TFTP and HTTP root directories and server addresses. For example:

```
[pxe]
# Ironic compute node's tftp root path. (string value)
tftp_root=/tftpboot

# IP address of Ironic compute node's tftp server. (string
# value)
tftp_server=192.168.0.2

[deploy]
# Ironic compute node's http root path. (string value)
http_root=/httpboot

# Ironic compute node's HTTP server URL. Example:
# http://192.1.2.3:8080 (string value)
http_url=http://192.168.0.2:8080
```

See also: *Deploying outside of the provisioning network.*

4. Install the iPXE package with the boot images:

Ubuntu:

```
apt-get install ipxe
```

RHEL8/CentOS8/Fedora:

```
dnf install ipxe-bootimgs
```

Note: SUSE does not provide a package containing iPXE boot images. If you are using SUSE or if the packaged version of the iPXE boot image doesn't work, you can download a prebuilt one

from <http://boot.ipxe.org> or build one image from source, see <http://ipxe.org/download> for more information.

- Copy the iPXE boot image (`undionly.kpxe` for **BIOS** and `ipxe.efi` for **UEFI**) to `/tftpboot`. The binary might be found at:

Ubuntu:

```
cp /usr/lib/ipxe/{undionly.kpxe,ipxe.efi,snponly.efi} /tftpboot
```

Fedora/RHEL8/CentOS8:

```
cp /usr/share/ipxe/{undionly.kpxe,ipxe.efi,snponly.efi} /tftpboot
```

- Enable/Configure iPXE overrides in the Bare Metal Services configuration file **if required** (`/etc/ironic/ironic.conf`):

```
[pxe]

# Neutron bootfile DHCP parameter. (string value)
ipxe_bootfile_name=undionly.kpxe

# Bootfile DHCP parameter for UEFI boot mode. (string value)
uefi_ipxe_bootfile_name=ipxe.efi

# Template file for PXE configuration. (string value)
ipxe_config_template=$pybasedir/drivers/modules/ipxe_config.template
```

Note: Most UEFI systems have integrated networking which means the `[pxe]uefi_ipxe_bootfile_name` setting should be set to `snponly.efi`.

Note: Setting the iPXE parameters noted in the code block above to no value, in other words setting a line to something like `ipxe_bootfile_name=` will result in ironic falling back to the default values of the non-iPXE PXE settings. This is for backwards compatibility.

- Ensure iPXE is the default PXE, if applicable.

In earlier versions of ironic, a `[pxe]ipxe_enabled` setting allowing operators to declare the behavior of the conductor to exclusively operate as if only iPXE was to be used. As time moved on, iPXE functionality was moved to its own `ipxe` boot interface.

If you want to emulate that same behavior, set the following in the configuration file (`/etc/ironic/ironic.conf`):

```
[DEFAULT]
default_boot_interface=ipxe
enabled_boot_interfaces=ipxe,pxe
```

Note: The `[DEFAULT]enabled_boot_interfaces` setting may be exclusively set to `ipxe`, how-

ever ironic has multiple interfaces available depending on the hardware types available for use.

8. It is possible to configure the Bare Metal service in such a way that nodes will boot into the deploy image directly from Object Storage. Doing this avoids having to cache the images on the ironic-conductor host and serving them via the ironic-conductors *HTTP server*. This can be done if:
 1. the Image Service is used for image storage;
 2. the images in the Image Service are internally stored in Object Storage;
 3. the Object Storage supports generating temporary URLs for accessing objects stored in it. Both the OpenStack Swift and RADOS Gateway provide support for this.
 - See *Ceph Object Gateway support* on how to configure the Bare Metal Service with RADOS Gateway as the Object Storage.

Configure this by setting the `[pxe]/ipxe_use_swift` configuration option to True as follows:

```
[pxe]
# Download deploy images directly from swift using temporary
# URLs. If set to false (default), images are downloaded to
# the ironic-conductor node and served over its local HTTP
# server. Applicable only when 'ipxe_enabled' option is set to
# true. (boolean value)
ipxe_use_swift=True
```

Although the *HTTP server* still has to be deployed and configured (as it will serve iPXE boot script and boot configuration files for nodes), such configuration will shift some load from ironic-conductor hosts to the Object Storage service which can be scaled horizontally.

Note that when SSL is enabled on the Object Storage service you have to ensure that iPXE firmware on the nodes can indeed boot from generated temporary URLs that use HTTPS protocol.

9. Restart the `ironic-conductor` process:

Fedora/RHEL8/CentOS8/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

PXE multi-architecture setup

It is possible to deploy servers of different architecture by one conductor. To use this feature, architecture-specific boot and template files must be configured using the configuration options `[pxe]pxe_bootfile_name_by_arch` and `[pxe]pxe_config_template_by_arch` respectively, in the Bare Metal services configuration file (`/etc/ironic/ironic.conf`).

These two options are dictionary values; the key is the architecture and the value is the boot (or config template) file. A nodes `cpu_arch` property is used as the key to get the appropriate boot file and template file. If the nodes `cpu_arch` is not in the dictionary, the configuration options

(in [pxe] group) `pxe_bootfile_name`, `pxe_config_template`, `uefi_pxe_bootfile_name` and `uefi_pxe_config_template` will be used instead.

In the following example, since `x86` and `x86_64` keys are not in the `pxe_bootfile_name_by_arch` or `pxe_config_template_by_arch` options, `x86` and `x86_64` nodes will be deployed by `pxelinux.0` or `bootx64.efi`, depending on the nodes `boot_mode` capability (`bios` or `uefi`). However, `aarch64` nodes will be deployed by `grubaa64.efi`, and `ppc64` nodes by `bootppc64`:

```
[pxe]

# Bootfile DHCP parameter. (string value)
pxe_bootfile_name=pxelinux.0

# On ironic-conductor node, template file for PXE
# configuration. (string value)
pxe_config_template = $pybasedir/drivers/modules/pxe_config.template

# Bootfile DHCP parameter for UEFI boot mode. (string value)
uefi_pxe_bootfile_name=bootx64.efi

# On ironic-conductor node, template file for PXE
# configuration for UEFI boot loader. (string value)
uefi_pxe_config_template=$pybasedir/drivers/modules/pxe_grub_config.template

# Bootfile DHCP parameter per node architecture. (dict value)
pxe_bootfile_name_by_arch=aarch64:grubaa64.efi,ppc64:bootppc64

# On ironic-conductor node, template file for PXE
# configuration per node architecture. For example:
# aarch64:/opt/share/grubaa64_pxe_config.template (dict value)
pxe_config_template_by_arch=aarch64:pxe_grubaa64_config.template,ppc64:pxe_
↪ppc64_config.template
```

Note: The grub implementation may vary on different architecture, you may need to tweak the pxe config template for a specific arch. For example, `grubaa64.efi` shipped with CentoOS7 does not support `linuxefi` and `initrdefi` commands, you'll need to switch to use `linux` and `initrd` command instead.

Note: A `[pxe]ipxe_bootfile_name_by_arch` setting is available for multi-arch iPXE based deployment, and defaults to the same behavior as the comparable `[pxe]pxe_bootfile_by_arch` setting for standard PXE.

PXE timeouts tuning

Because of its reliance on UDP-based protocols (DHCP and TFTP), PXE is particularly vulnerable to random failures during the booting stage. If the deployment ramdisk never calls back to the bare metal conductor, the build will be aborted, and the node will be moved to the `deploy failed` state, after the `deploy callback timeout`. This timeout can be changed via the `conductor.deploy_callback_timeout` configuration option.

Starting with the Train release, the Bare Metal service can retry PXE boot if it takes too long. The timeout is defined via `pxe.boot_retry_timeout` and must be smaller than the `deploy_callback_timeout`, otherwise it will have no effect.

For example, the following configuration sets the overall timeout to 60 minutes, allowing two retries after 20 minutes:

```
[conductor]
deploy_callback_timeout = 3600

[pxe]
boot_retry_timeout = 1200
```

Configuring IPMI support

Installing ipmitool command

To enable one of the drivers that use IPMI protocol for power and management actions (for example, `ipmi`), the `ipmitool` command must be present on the service node(s) where `ironic-conductor` is running. On most distros, it is provided as part of the `ipmitool` package. Source code is available at <http://ipmitool.sourceforge.net/>.

Warning: Certain distros, notably Mac OS X and SLES, install `openipmi` instead of `ipmitool` by default. This driver is not compatible with `openipmi` as it relies on error handling options not provided by this tool.

Please refer to the *IPMI driver* for information on how to use IPMItool-based drivers.

Validation and troubleshooting

Check that you can connect to, and authenticate with, the IPMI controller in your bare metal server by running `ipmitool`:

```
ipmitool -I lanplus -H <ip-address> -U <username> -P <password> chassis power_↵
↵status
```

where `<ip-address>` is the IP of the IPMI controller you want to access. This is not the bare metal nodes main IP. The IPMI controller should have its own unique IP.

If the above command doesnt return the power status of the bare metal server, check that

- `ipmitool` is installed and is available via the `$PATH` environment variable.

- The IPMI controller on your bare metal server is turned on.
- The IPMI controller credentials and IP address passed in the command are correct.
- The conductor node has a route to the IPMI controller. This can be checked by just pinging the IPMI controller IP from the conductor node.

IPMI configuration

If there are slow or unresponsive BMCs in the environment, the `min_command_interval` configuration option in the `[ipmi]` section may need to be raised. The default is fairly conservative, as setting this timeout too low can cause older BMCs to crash and require a hard-reset.

Collecting sensor data

Bare Metal service supports sending IPMI sensor data to Telemetry with certain hardware types, such as `ipmi`, `ilo` and `irmc`. By default, support for sending IPMI sensor data to Telemetry is disabled. If you want to enable it, you should make the following two changes in `ironic.conf`:

```
[conductor]
send_sensor_data = true
[oslo_messaging_notifications]
driver = messagingv2
```

If you want to customize the sensor types which will be sent to Telemetry, change the `send_sensor_data_types` option. For example, the below settings will send information about temperature, fan, voltage from sensors to the Telemetry service:

```
send_sensor_data_types=Temperature,Fan,Voltage
```

Supported sensor types are defined by the Telemetry service, currently these are `Temperature`, `Fan`, `Voltage`, `Current`. Special value `All` (the default) designates all supported sensor types.

2.1.7 Enrollment

After all the services have been properly configured, you should enroll your hardware with the Bare Metal service, and confirm that the Compute service sees the available hardware. The nodes will be visible to the Compute service once they are in the `available` provision state.

Note: After enrolling nodes with the Bare Metal service, the Compute service will not be immediately notified of the new resources. The Compute services resource tracker syncs periodically, and so any changes made directly to the Bare Metal services resources will become visible in the Compute service only after the next run of that periodic task. More information is in the [Troubleshooting](#) section.

Note: Any bare metal node that is visible to the Compute service may have a workload scheduled to it, if both the `power` and `management` interfaces pass the `validate` check. If you wish to exclude a node

from the Compute services scheduler, for instance so that you can perform maintenance on it, you can set the node to maintenance mode. For more information see the *Maintenance mode* section.

Choosing a driver

When enrolling a node, the most important information to supply is *driver*. See *Enabling drivers and hardware types* for a detailed explanation of bare metal drivers, hardware types and interfaces. The `driver list` command can be used to list all drivers enabled on all hosts:

```
baremetal driver list
+-----+-----+
| Supported driver(s) | Active host(s)      |
+-----+-----+
| ipmi                | localhost.localdomain |
+-----+-----+
```

The specific driver to use should be picked based on actual hardware capabilities and expected features. See *Drivers, Hardware Types and Hardware Interfaces* for more hints on that.

Each driver has a list of *driver properties* that need to be specified via the nodes `driver_info` field, in order for the driver to operate on node. This list consists of the properties of the hardware interfaces that the driver uses. These driver properties are available with the `driver property list` command:

```
$ baremetal driver property list ipmi
+-----+-----+
↪-----+
| Property          | Description                                     |
↪-----+
+-----+-----+
↪-----+
| ipmi_address      | IP address or hostname of the node. Required. |
↪-----+
| ipmi_password     | password. Optional.                            |
↪-----+
| ipmi_username     | username; default is NULL user. Optional.     |
↪-----+
| ...               | ...                                             |
↪-----+
| deploy_kernel     | UUID (from Glance) of the deployment kernel.↪
↪Required.          |
| deploy_ramdisk    | UUID (from Glance) of the ramdisk that is mounted at↪
↪boot time. Required. |
+-----+-----+
↪-----+
```

The properties marked as required must be supplied either during node creation or shortly after. Some properties may only be required for certain features.

Note on API versions

Starting with API version 1.11, the Bare Metal service added a new initial provision state of `enroll` to its state machine. When this or later API version is used, new nodes get this state instead of `available`.

Existing automation tooling that use an API version lower than 1.11 are not affected, since the initial provision state is still `available`. However, using API version 1.11 or above may break existing automation tooling with respect to node creation.

The default API version used by (the most recent) `python-ironicclient` is 1.9, but it may change in the future and should not be relied on.

In the examples below we will use version 1.11 of the Bare metal API. This gives us the following advantages:

- Explicit power credentials validation before leaving the `enroll` state.
- Running node cleaning before entering the `available` state.
- Not exposing half-configured nodes to the scheduler.

To set the API version for all commands, you can set the environment variable `IRONIC_API_VERSION`. For the OpenStackClient baremetal plugin, set the `OS_BAREMETAL_API_VERSION` variable to the same value. For example:

```
$ export IRONIC_API_VERSION=1.11
$ export OS_BAREMETAL_API_VERSION=1.11
```

Enrollment process

Creating a node

This section describes the main steps to enroll a node and make it available for provisioning. Some steps are shown separately for illustration purposes, and may be combined if desired.

1. Create a node in the Bare Metal service with the `node create` command. At a minimum, you must specify the driver name (for example, `ipmi`).

This command returns the node UUID along with other information about the node. The nodes provision state will be `enroll`:

```
$ export OS_BAREMETAL_API_VERSION=1.11
$ baremetal node create --driver ipmi
+-----+-----+
| Property      | Value                                |
+-----+-----+
| uuid          | dfc6189f-ad83-4261-9bda-b27258eb1987 |
| driver_info   | {}                                    |
| extra         | {}                                    |
| driver        | ipmi                                  |
| chassis_uuid  |                                         |
| properties    | {}                                    |
| name          | None                                  |
+-----+-----+
```

(continues on next page)

(continued from previous page)

```

$ baremetal node show dfc6189f-ad83-4261-9bda-b27258eb1987
+-----+-----+
| Property          | Value          |
+-----+-----+
| target_power_state | None          |
| extra             | {}            |
| last_error        | None          |
| maintenance_reason | None          |
| provision_state    | enroll        |
| uuid              | dfc6189f-ad83-4261-9bda-b27258eb1987 |
| console_enabled    | False         |
| target_provision_state | None          |
| provision_updated_at | None          |
| maintenance        | False         |
| power_state        | None          |
| driver             | ipmi          |
| properties         | {}            |
| instance_uuid      | None          |
| name               | None          |
| driver_info        | {}            |
| ...                | ...           |
+-----+-----+

```

A node may also be referred to by a logical name as well as its UUID. A name can be assigned to the node during its creation by adding the `-n` option to the `node create` command or by updating an existing node with the `node set` command. See *Logical Names* for examples.

- Starting with API version 1.31 (and `python-ironicclient` 1.13), you can pick which hardware interface to use with nodes that use hardware types. Each interface is represented by a node field called `<IFACE>_interface` where `<IFACE>` is the interface type, e.g. `boot`. See *Enabling drivers and hardware types* for details on hardware interfaces.

An interface can be set either separately:

```

$ baremetal node set $NODE_UUID --deploy-interface direct --raid-
↵interface agent

```

or set during node creation:

```

$ baremetal node create --driver ipmi \
  --deploy-interface direct \
  --raid-interface agent

```

If no value is provided for some interfaces, *Defaults for hardware interfaces* are used instead.

- Update the node `driver_info` with the required driver properties, so that the Bare Metal service can manage the node:

```

$ baremetal node set $NODE_UUID \
  --driver-info ipmi_username=$USER \

```

(continues on next page)

(continued from previous page)

```
--driver-info ipmi_password=$PASS \  
--driver-info ipmi_address=$ADDRESS
```

Note: If IPMI is running on a port other than 623 (the default). The port must be added to `driver_info` by specifying the `ipmi_port` value. Example:

```
$ baremetal node set $NODE_UUID --driver-info ipmi_port=$PORT_NUMBER
```

You may also specify all `driver_info` parameters during node creation by passing the **driver-info** option multiple times:

```
$ baremetal node create --driver ipmi \  
--driver-info ipmi_username=$USER \  
--driver-info ipmi_password=$PASS \  
--driver-info ipmi_address=$ADDRESS
```

See *Choosing a driver* above for details on driver properties.

- Specify a deploy kernel and ramdisk compatible with the nodes driver, for example:

```
$ baremetal node set $NODE_UUID \  
--driver-info deploy_kernel=$DEPLOY_VMLINUZ_UUID \  
--driver-info deploy_ramdisk=$DEPLOY_INITRD_UUID
```

See *Add images to the Image service* for details.

- Optionally you can specify the provisioning and/or cleaning network UUID or name in the nodes `driver_info`. The neutron network interface requires both `provisioning_network` and `cleaning_network`, while the flat network interface requires the `cleaning_network` to be set either in the configuration or on the nodes. For example:

```
$ baremetal node set $NODE_UUID \  
--driver-info cleaning_network=$CLEAN_UUID_OR_NAME \  
--driver-info provisioning_network=$PROVISION_UUID_OR_NAME
```

See *Configure tenant networks* for details.

- You must also inform the Bare Metal service of the network interface cards which are part of the node by creating a port with each NICs MAC address. These MAC addresses are passed to the Networking service during instance provisioning and used to configure the network appropriately:

```
$ baremetal port create $MAC_ADDRESS --node $NODE_UUID
```

Note: When it is time to remove the node from the Bare Metal service, the command used to remove the port is `baremetal port delete <port uuid>`. When doing so, it is important to ensure that the baremetal node is not in maintenance as guarding logic to prevent orphaning Neutron Virtual Interfaces (VIFs) will be overridden.

Adding scheduling information

1. Assign a *resource class* to the node. A *resource class* should represent a class of hardware in your data center, that corresponds to a Compute flavor.

For example, lets split hardware into these three groups:

1. nodes with a lot of RAM and powerful CPU for computational tasks,
2. nodes with powerful GPU for OpenCL computing,
3. smaller nodes for development and testing.

We can define three resource classes to reflect these hardware groups, named `large-cpu`, `large-gpu` and `small` respectively. Then, for each node in each of the hardware groups, well set their `resource_class` appropriately via:

```
$ baremetal node set $NODE_UUID --resource-class $CLASS_NAME
```

The `--resource-class` argument can also be used when creating a node:

```
$ baremetal node create --driver $DRIVER --resource-class $CLASS_NAME
```

To use resource classes for scheduling you need to update your flavors as described in *Create flavors for use with the Bare Metal service*.

Note: This is not required for standalone deployments, only for those using the Compute service for provisioning bare metal instances.

2. Update the nodes properties to match the actual hardware of the node:

```
$ baremetal node set $NODE_UUID \  
  --property cpus=$CPU_COUNT \  
  --property memory_mb=$RAM_MB \  
  --property local_gb=$DISK_GB
```

As above, these can also be specified at node creation by passing the **property** option to node create multiple times:

```
$ baremetal node create --driver ipmi \  
  --driver-info ipmi_username=$USER \  
  --driver-info ipmi_password=$PASS \  
  --driver-info ipmi_address=$ADDRESS \  
  --property cpus=$CPU_COUNT \  
  --property memory_mb=$RAM_MB \  
  --property local_gb=$DISK_GB
```

These values can also be discovered during *Hardware Inspection*.

Warning: The value provided for the `local_gb` property must match the size of the root device youre going to deploy on. By default **ironic-python-agent** picks the smallest disk which is not smaller than 4 GiB.

If you override this logic by using root device hints (see *Specifying the disk for deployment (root device hints)*), the `local_gb` value should match the size of picked target disk.

- If you wish to perform more advanced scheduling of the instances based on hardware capabilities, you may add metadata to each node that will be exposed to the Compute scheduler (see: [ComputeCapabilitiesFilter](#)). A full explanation of this is outside of the scope of this document. It can be done through the special `capabilities` member of node properties:

```
$ baremetal node set $NODE_UUID \
  --property capabilities=key1:val1,key2:val2
```

Some capabilities can also be discovered during *Hardware Inspection*.

- If you wish to perform advanced scheduling of instances based on qualitative attributes of bare metal nodes, you may add traits to each bare metal node that will be exposed to the Compute scheduler (see: *Scheduling based on traits* for a more in-depth discussion of traits in the Bare Metal service). For example, to add the standard trait `HW_CPU_X86_VMX` and a custom trait `CUSTOM_TRAIT1` to a node:

```
$ baremetal node add trait $NODE_UUID \
  CUSTOM_TRAIT1 HW_CPU_X86_VMX
```

Validating node information

- To check if Bare Metal service has the minimum information necessary for a nodes driver to be functional, you may validate it:

```
$ baremetal node validate $NODE_UUID
+-----+-----+-----+
| Interface | Result | Reason |
+-----+-----+-----+
| boot      | True   |         |
| console   | True   |         |
| deploy    | True   |         |
| inspect   | True   |         |
| management | True   |         |
| network   | True   |         |
| power     | True   |         |
| raid      | True   |         |
| storage   | True   |         |
+-----+-----+-----+
```

If the node fails validation, each driver interface will return information as to why it failed:

```
$ baremetal node validate $NODE_UUID
+-----+-----+-----+
↪ -----
↪ -----
↪ -----+
| Interface | Result | Reason |
↪ -----
↪ -----
```

(continues on next page)

(continued from previous page)

```

+-----+-----+-----+
↪-----+
↪-----+
| boot      | True   |
↪
↪
| console   | None   | not supported
↪
↪
| deploy    | False  | Cannot validate iSCSI deploy. Some parameters
↪were missing in node's instance_info. Missing are: ['root_gb', 'image_
↪source']
| inspect   | True   |
↪
↪
| management | False  | Missing the following IPMI credentials in node's
↪driver_info: ['ipmi_address'].
↪
↪
| network   | True   |
↪
↪
| power     | False  | Missing the following IPMI credentials in node's
↪driver_info: ['ipmi_address'].
↪
↪
| raid      | None   | not supported
↪
↪
| storage   | True   |
↪
↪
+-----+-----+-----+
↪-----+
↪-----+

```

When using the Compute Service with the Bare Metal service, it is safe to ignore the deploy interfaces validation error due to lack of image information. You may continue the enrollment process. This information will be set by the Compute Service just before deploying, when an instance is requested:

```

$ baremetal node validate $NODE_UUID
+-----+-----+-----+
↪-----+
↪-----+
| Interface | Result | Reason
↪
↪
↪
+-----+-----+-----+
↪-----+
↪-----+
↪-----+

```

(continues on next page)

(continued from previous page)

```

| boot          | False | Cannot validate image information for node
↳because one or more parameters are missing from its instance_info.
↳Missing are: ['ramdisk', 'kernel', 'image_source'] |
| console       | True  |
↳
↳
| deploy        | False | Cannot validate image information for node
↳because one or more parameters are missing from its instance_info.
↳Missing are: ['ramdisk', 'kernel', 'image_source'] |
| inspect       | True  |
↳
↳
| management    | True  |
↳
↳
| network       | True  |
↳
↳
| power         | True  |
↳
↳
| raid          | None  | not supported
↳
↳
| storage       | True  |
↳
↳
+-----+-----+-----+-----+
↳-----+
↳-----+

```

Making node available for deployment

In order for nodes to be available for deploying workloads on them, nodes must be in the available provision state. To do this, nodes created with API version 1.11 and above must be moved from the enroll state to the manageable state and then to the available state. This section can be safely skipped, if API version 1.10 or earlier is used (which is the case by default).

After creating a node and before moving it from its initial provision state of `enroll`, basic power and port information needs to be configured on the node. The Bare Metal service needs this information because it verifies that it is capable of controlling the node when transitioning the node from `enroll` to manageable state.

To move a node from `enroll` to manageable provision state:

```

$ baremetal node manage $NODE_UUID
$ baremetal node show $NODE_UUID
+-----+-----+-----+-----+
↳-----+

```

(continues on next page)

(continued from previous page)

Property	Value
...	...
provision_state	manageable
uuid	0eb013bb-1e4b-4f4c-94b5-2e7468242611
...	...

Note: Since it is an asynchronous call, the response for `baremetal node manage` will not indicate whether the transition succeeded or not. You can check the status of the operation via `baremetal node show`. If it was successful, `provision_state` will be in the desired state. If it failed, there will be information in the nodes `last_error`.

When a node is moved from the `manageable` to `available` provision state, the node will go through automated cleaning if configured to do so (see *Configure the Bare Metal service for cleaning*).

To move a node from `manageable` to `available` provision state:

```
$ baremetal node provide $NODE_UUID
$ baremetal node show $NODE_UUID
```

Property	Value
...	...
provision_state	available
uuid	0eb013bb-1e4b-4f4c-94b5-2e7468242611
...	...

For more details on the Bare Metal services state machine, see the *Ironics State Machine* documentation.

Mapping nodes to Compute cells

If the Compute service is used for scheduling, and the `discover_hosts_in_cells_interval` was not set as described in *Configure the Compute service to use the Bare Metal service*, then log into any controller node and run the following command to map the new node(s) to Compute cells:

```
nova-manage cell_v2 discover_hosts
```

Logical names

A node may also be referred to by a logical name as well as its UUID. Names can be assigned either during its creation by adding the `-n` option to the `node create` command or by updating an existing node with the `node set` command.

Node names must be unique, and conform to:

- `rfc952`
- `rfc1123`
- `wiki_hostname`

The node is named `example` in the following examples:

```
$ baremetal node create --driver ipmi --name example
```

or

```
$ baremetal node set $NODE_UUID --name example
```

Once assigned a logical name, a node can then be referred to by name or UUID interchangeably:

```
$ baremetal node create --driver ipmi --name example
+-----+-----+
| Property      | Value                               |
+-----+-----+
| uuid          | 71e01002-8662-434d-aafd-f068f69bb85e |
| driver_info   | {}                                   |
| extra         | {}                                   |
| driver        | ipmi                                 |
| chassis_uuid  |                                       |
| properties    | {}                                   |
| name          | example                              |
+-----+-----+

$ baremetal node show example
+-----+-----+
| Property      | Value                               |
+-----+-----+
| target_power_state | None                                 |
| extra         | {}                                   |
| last_error      | None                                 |
+-----+-----+
```

(continues on next page)

(continued from previous page)

updated_at	2015-04-24T16:23:46+00:00	
...	...	
instance_info	{}	
+-----+-----+		

Defaults for hardware interfaces

For *hardware types*, users can request one of enabled implementations when creating or updating a node as explained in *Creating a node*.

When no value is provided for a certain interface when creating a node, or changing a nodes hardware type, the default value is used. You can use the driver details command to list the current enabled and default interfaces for a hardware type (for your deployment):

```
$ baremetal driver show ipmi
```

Field	Value
default_boot_interface	pxe
default_console_interface	no-console
default_deploy_interface	direct
default_inspect_interface	no-inspect
default_management_interface	ipmitool
default_network_interface	flat
default_power_interface	ipmitool
default_raid_interface	no-raid
default_vendor_interface	no-vendor
enabled_boot_interfaces	pxe
enabled_console_interfaces	no-console
enabled_deploy_interfaces	direct
enabled_inspect_interfaces	no-inspect
enabled_management_interfaces	ipmitool
enabled_network_interfaces	flat, noop
enabled_power_interfaces	ipmitool
enabled_raid_interfaces	no-raid, agent
enabled_vendor_interfaces	no-vendor
hosts	ironic-host-1
name	ipmi
type	dynamic

The defaults are calculated as follows:

1. If the `default_<IFACE>_interface` configuration option (where `<IFACE>` is the interface name) is set, its value is used as the default.

If this implementation is not compatible with the nodes hardware type, an error is returned to a user. An explicit value has to be provided for the nodes `<IFACE>_interface` field in this case.

2. Otherwise, the first supported implementation that is enabled by an operator is used as the default.

A list of supported implementations is calculated by taking the intersection between the implementations supported by the nodes hardware type and implementations enabled by the `enabled_<IFACE>_interfaces` option (where <IFACE> is the interface name). The calculation preserves the order of items, as provided by the hardware type.

If the list of supported implementations is not empty, the first one is used. Otherwise, an error is returned to a user. In this case, an explicit value has to be provided for the <IFACE>_interface field.

See *Enabling drivers and hardware types* for more details on configuration.

Example

Consider the following configuration (shortened for simplicity):

```
[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_console_interfaces = no-console,ipmitool-shellinabox
enabled_deploy_interfaces = direct
enabled_management_interfaces = ipmitool,redfish
enabled_power_interfaces = ipmitool,redfish
default_deploy_interface = ansible
```

A new node is created with the `ipmi` driver and no interfaces specified:

```
$ export OS_BAREMETAL_API_VERSION=1.31
$ baremetal node create --driver ipmi
+-----+-----+
| Property | Value |
+-----+-----+
| uuid     | dfc6189f-ad83-4261-9bda-b27258eb1987 |
| driver_info | {} |
| extra     | {} |
| driver    | ipmi |
| chassis_uuid | |
| properties | {} |
| name     | None |
+-----+-----+
```

Then the defaults for the interfaces that will be used by the node in this example are calculated as follows:

deploy An explicit value of `ansible` is provided for `default_deploy_interface`, so it is used.

power No default is configured. The `ipmi` hardware type supports only `ipmitool` power. The intersection between supported power interfaces and values provided in the `enabled_power_interfaces` option has only one item: `ipmitool`. It is used.

console No default is configured. The `ipmi` hardware type supports the following console interfaces: `ipmitool-socat`, `ipmitool-shellinabox` and `no-console` (in this order). Of these three, only two are enabled: `no-console` and `ipmitool-shellinabox` (order does not matter). The intersection contains `ipmitool-shellinabox` and `no-console`. The first item is used, and it is `ipmitool-shellinabox`.

management Following the same calculation as *power*, the `ipmitool` management interface is used.

Hardware Inspection

The Bare Metal service supports hardware inspection that simplifies enrolling nodes - please see *Hardware Inspection* for details.

Tenant Networks and Port Groups

See *Multi-tenancy in the Bare Metal service* and *Port groups support*.

2.1.8 Using Bare Metal service as a standalone service

This guide explains how to configure and use the Bare Metal service standalone, i.e. without other OpenStack services. In this mode users are interacting with the bare metal API directly, not through OpenStack Compute.

Configuration

This guide covers manual configuration of the Bare Metal service in the standalone mode. Alternatively, *Bifrost* can be used for automatic configuration.

Service settings

It is possible to use the Bare Metal service without other OpenStack services. You should make the following changes to `/etc/ironic/ironic.conf`:

1. Choose an authentication strategy which supports standalone, one option is `noauth`:

```
[DEFAULT]
auth_strategy=noauth
```

Another option is `http_basic` where the credentials are stored in an *Apache htpasswd* format file:

```
[DEFAULT]
auth_strategy=http_basic
http_basic_auth_user_file=/etc/ironic/htpasswd
```

Only the `bcrypt` format is supported, and the *Apache htpasswd* utility can be used to populate the file with entries, for example:

```
htpasswd -nbB myName myPassword >> /etc/ironic/htpasswd
```

2. If you want to disable the Networking service, you should have your network pre-configured to serve DHCP and TFTP for machines that you're deploying. To disable it, change the following lines:

```
[dhcp]
dhcp_provider=none
```

Note: If you disabled the Networking service and the driver that you use is supported by at most one conductor, PXE boot will still work for your nodes without any manual config editing. This is because you know all the DHCP options that will be used for deployment and can set up your DHCP server appropriately.

If you have multiple conductors per driver, it would be better to use Networking since it will do all the dynamically changing configurations for you.

3. If you want to disable using a messaging broker between conductor and API processes, switch to JSON RPC instead:

```
[DEFAULT]
rpc_transport = json-rpc
```

JSON RPC also has its own authentication strategy. If it is not specified then the strategy defaults to [DEFAULT] `auth_strategy`. The following will set JSON RPC to `noauth`:

```
[json_rpc]
auth_strategy = noauth
```

For `http_basic` the conductor server needs a credentials file to validate requests:

```
[json_rpc]
auth_strategy = http_basic
http_basic_auth_user_file = /etc/ironic/htpasswd-json-rpc
```

The API server also needs client-side credentials to be specified:

```
[json_rpc]
auth_type = http_basic
username = myName
password = myPassword
```

Using CLI

To use the `baremetal CLI`, set up these environment variables. If the `noauth` authentication strategy is being used, the value `none` must be set for `OS_AUTH_TYPE`. `OS_ENDPOINT` is the URL of the `ironic-api` process. For example:

```
export OS_AUTH_TYPE=none
export OS_ENDPOINT=http://localhost:6385/
```

If the `http_basic` authentication strategy is being used, the value `http_basic` must be set for `OS_AUTH_TYPE`. For example:

```
export OS_AUTH_TYPE=http_basic
export OS_ENDPOINT=http://localhost:6385/
export OS_USERNAME=myUser
export OS_PASSWORD=myPassword
```

Enrollment

Preparing images

If you don't use Image service, it's possible to provide images to Bare Metal service via a URL.

At the moment, only two types of URLs are acceptable instead of Image service UUIDs: HTTP(S) URLs (for example, <http://my.server.net/images/img>) and file URLs (<file:///images/img>).

There are however some limitations for different hardware interfaces:

- If you're using *Direct deploy* with HTTP(s) URLs, you have to provide the Bare Metal service with the a checksum of your instance image.

MD5 is used by default for backward compatibility reasons. To compute an MD5 checksum, you can use the following command:

```
$ md5sum image.qcow2
ed82def8730f394fb85aef8a208635f6 image.qcow2
```

Alternatively, use a SHA256 checksum or any other algorithm supported by the Python's `hashlib`, e.g.:

```
$ sha256sum image.qcow2
9f6c942ad81690a9926ff530629fb69a82db8b8ab267e2cbd59df417c1a28060 image.
↪qcow2
```

- *Direct deploy* started supporting `file://` images in the Victoria release cycle, before that only HTTP(s) had been supported.

Warning: File images must be accessible to every conductor! Use a shared file system if you have more than one conductor. The ironic CLI tool will not transfer the file from a local machine to the conductor(s).

Note: The Bare Metal service tracks content changes for non-Glance images by checking their modification date and time. For example, for HTTP image, if Last-Modified header value from response to a HEAD request to <http://my.server.net/images/deploy.ramdisk> is greater than cached image modification time, Ironic will re-download the content. For `file://` images, the file system modification time is used.

Enrolling nodes

1. Create a node in Bare Metal service. At minimum, you must specify the driver name (for example, `ipmi`). You can also specify all the required driver parameters in one command. This will return the node UUID:

```
$ baremetal node create --driver ipmi \
  --driver-info ipmi_address=ipmi.server.net \
  --driver-info ipmi_username=user \
  --driver-info ipmi_password=pass \
```

(continues on next page)

(continued from previous page)

```

--driver-info deploy_kernel=file:///images/deploy.vmlinuz \
--driver-info deploy_ramdisk=http://my.server.net/images/deploy.
↪ ramdisk
+-----+
↪ -----+
| Property      | Value
↪
+-----+
↪ -----+
| uuid          | be94df40-b80a-4f63-b92b-e9368ee8d14c
↪
| driver_info   | {u'deploy_ramdisk': u'http://my.server.net/images/deploy.
↪ ramdisk',
|               | u'deploy_kernel': u'file:///images/deploy.vmlinuz', u
↪ 'ipmi_address':
|               | u'ipmi.server.net', u'ipmi_username': u'user', u'ipmi_
↪ password':
|               | u'*****'}
↪
| extra         | {}
↪
| driver        | ipmi
↪
| chassis_uuid |
↪
| properties    | {}
↪
+-----+
↪ -----+

```

Note that here `deploy_kernel` and `deploy_ramdisk` contain links to images instead of Image service UUIDs.

2. As in case of Compute service, you can also provide `capabilities` to node properties, but they will be used only by Bare Metal service (for example, boot mode). Although you don't need to add properties like `memory_mb`, `cpus` etc. as Bare Metal service will require UUID of a node you're going to deploy.
3. Then create a port to inform Bare Metal service of the network interface cards which are part of the node by creating a port with each NIC's MAC address. In this case, they're used for naming of PXE configs for a node:

```
baremetal port create $MAC_ADDRESS --node $NODE_UUID
```

Once the installation is done, please see *Deploying with Bare Metal service* for information on how to deploy bare metal machines.

Deploying

The content has been migrated, please see *Deploying with Bare Metal service*.

2.1.9 Enabling the configuration drive (configdrive)

The Bare Metal service supports exposing a configuration drive image to the instances.

The configuration drive is used to store instance-specific metadata and is present to the instance as a disk partition labeled `config-2`. The configuration drive has a maximum size of 64MB. One use case for using the configuration drive is to expose a networking configuration when you do not use DHCP to assign IP addresses to instances.

The configuration drive is usually used in conjunction with the Compute service, but the Bare Metal service also offers a standalone way of using it. The following sections will describe both methods.

When used with Compute service

To enable the configuration drive for a specific request, pass `--config-drive true` parameter to the `nova boot` command, for example:

```
nova boot --config-drive true --flavor baremetal --image test-image instance-1
```

Its also possible to enable the configuration drive automatically on all instances by configuring the OpenStack Compute service to always create a configuration drive by setting the following option in the `/etc/nova/nova.conf` file, for example:

```
[DEFAULT]
...

force_config_drive=True
```

In some cases, you may wish to pass a user customized script when deploying an instance. To do this, pass `--user-data /path/to/file` to the `nova boot` command.

When used standalone

When used without the Compute service, the operator needs to create a configuration drive and provide the file or HTTP URL to the Bare Metal service. See *Deploying with a config drive* for details.

Configuration drive storage in an object store

Under normal circumstances, the configuration drive can be stored in the Bare Metal service when the size is less than 64KB. Optionally, if the size is larger than 64KB there is support to store it in a swift endpoint. Both swift and radosgw use swift-style APIs.

The following option in `/etc/ironic/ironic.conf` enables swift as an object store backend to store config drive. This uses the Identity service to establish a session between the Bare Metal service and the Object Storage service.


```
[deploy]
...

configdrive_use_object_store = True
```

Use the following options in `/etc/ironic/ironic.conf` to enable radosgw. Credentials in the swift section are needed because radosgw will not use the Identity service and relies on radosgws username and password authentication instead.

```
[deploy]
...

configdrive_use_object_store = True

[swift]
...

username = USERNAME
password = PASSWORD
auth_url = http://RADOSGW_IP:8000/auth/v1
```

If the *Direct deploy* is being used, edit `/etc/glance/glance-api.conf` to store the instance images in respective object store (radosgw or swift) as well:

```
[glance_store]
...

swift_store_user = USERNAME
swift_store_key = PASSWORD
swift_store_auth_address = http://RADOSGW_OR_SWIFT_IP:PORT/auth/v1
```

Accessing the configuration drive data

When the configuration drive is enabled, the Bare Metal service will create a partition on the instance disk and write the configuration drive image onto it. The configuration drive must be mounted before use. This is performed automatically by many tools, such as cloud-init and cloudbase-init. To mount it manually on a Linux distribution that supports accessing devices by labels, simply run the following:

```
mkdir -p /mnt/config
mount /dev/disk/by-label/config-2 /mnt/config
```

If the guest OS doesn't support accessing devices by labels, you can use other tools such as `blkid` to identify which device corresponds to the configuration drive and mount it, for example:

```
CONFIG_DEV=$(blkid -t LABEL="config-2" -o device)
mkdir -p /mnt/config
mount $CONFIG_DEV /mnt/config
```

Cloud-init integration

The configuration drive can be especially useful when used with `cloud-init`, but in order to use it we should follow some rules:

- `Cloud-init` data should be organized in the [expected format](#).
- Since the Bare Metal service uses a disk partition as the configuration drive, it will only work with `cloud-init` version `>= 0.7.5`.
- `Cloud-init` has a collection of data source modules, so when building the image with `disk-image-builder` we have to define `DIB_CLOUD_INIT_DATASOURCES` environment variable and set the appropriate sources to enable the configuration drive, for example:

```
DIB_CLOUD_INIT_DATASOURCES="ConfigDrive, OpenStack" disk-image-create -o ↵  
↵ fedora-cloud-image fedora baremetal
```

For more information see [how to configure cloud-init data sources](#).

2.1.10 Advanced features

Local boot with partition images

The Bare Metal service supports local boot with partition images, meaning that after the deployment the nodes subsequent reboots won't happen via PXE or Virtual Media. Instead, it will boot from a local boot loader installed on the disk.

Note: Whole disk images, on the contrary, support only local boot, and use it by default.

It's important to note that in order for this to work the image being deployed with Bare Metal service **must** contain `grub2` installed within it.

Enabling the local boot is different when Bare Metal service is used with Compute service and without it. The following sections will describe both methods.

Enabling local boot with Compute service

To enable local boot we need to set a capability on the bare metal node, for example:

```
baremetal node set <node-uuid> --property capabilities="boot_option:local"
```

Nodes having `boot_option` set to `local` may be requested by adding an `extra_spec` to the Compute service flavor, for example:

```
nova flavor-key baremetal set capabilities:boot_option="local"
```

Note: If the node is configured to use UEFI, Bare Metal service will create an `EFI` partition on the disk and switch the partition table format to `gpt`. The `EFI` partition will be used later by the boot loader (which is installed from the `deploy ramdisk`).

Enabling local boot without Compute

Since adding capabilities to the nodes properties is only used by the nova scheduler to perform more advanced scheduling of instances, we need a way to enable local boot when Compute is not present. To do that we can simply specify the capability via the `instance_info` attribute of the node, for example:

```
baremetal node set <node-uuid> --instance-info capabilities='{"boot_option":
↳"local"}'
```

Specifying the disk for deployment (root device hints)

The Bare Metal service supports passing hints to the deploy ramdisk about which disk it should pick for the deployment. The list of supported hints is:

- `model` (STRING): device identifier
- `vendor` (STRING): device vendor
- `serial` (STRING): disk serial number
- `size` (INT): size of the device in GiB

Note: A nodes `local_gb` property is often set to a value 1 GiB less than the actual disk size to account for partitioning (this is how DevStack, TripleO and Ironic Inspector work, to name a few). However, in this case `size` should be the actual size. For example, for a 128 GiB disk `local_gb` will be 127, but `size` hint will be 128.

- `wwn` (STRING): unique storage identifier
- `wwn_with_extension` (STRING): unique storage identifier with the vendor extension appended
- `wwn_vendor_extension` (STRING): unique vendor storage identifier
- `rotational` (BOOLEAN): whether its a rotational device or not. This hint makes it easier to distinguish HDDs (rotational) and SSDs (not rotational) when choosing which disk Ironic should deploy the image onto.
- `hctl` (STRING): the SCSI address (Host, Channel, Target and Lun), e.g 1:0:0:0
- `by_path` (STRING): the alternate device name corresponding to a particular PCI or iSCSI path, e.g `/dev/disk/by-path/pci-0000:00`
- `name` (STRING): the device name, e.g `/dev/md0`

Warning: The root device hint name should only be used for devices with constant names (e.g RAID volumes). For SATA, SCSI and IDE disk controllers this hint is not recommended because the order in which the device nodes are added in Linux is arbitrary, resulting in devices like `/dev/sda` and `/dev/sdb` **switching around at boot time**.

To associate one or more hints with a node, update the nodes properties with a `root_device` key, for example:

```
baremetal node set <node-uuid> --property root_device='{"wwn":  
↳"0x4000cca77fc4dba1"}'
```

That will guarantee that Bare Metal service will pick the disk device that has the wwn equal to the specified wwn value, or fail the deployment if it can not be found.

Note: Starting with the Ussuri release, root device hints can be specified per-instance, see *Using Bare Metal service as a standalone service*.

The hints can have an operator at the beginning of the value string. If no operator is specified the default is == (for numerical values) and s== (for string values). The supported operators are:

- For numerical values:
 - = equal to or greater than. This is equivalent to >= and is supported for legacy reasons
 - == equal to
 - != not equal to
 - >= greater than or equal to
 - > greater than
 - <= less than or equal to
 - < less than
- For strings (as python comparisons):
 - s== equal to
 - s!= not equal to
 - s>= greater than or equal to
 - s> greater than
 - s<= less than or equal to
 - s< less than
 - <in> substring
- For collections:
 - <all-in> all elements contained in collection
 - <or> find one of these

Examples are:

- Finding a disk larger or equal to 60 GiB and non-rotational (SSD):

```
baremetal node set <node-uuid> --property root_device='{"size": ">= 60",  
↳"rotational": false}'
```

- Finding a disk whose vendor is samsung or winsys:

```
baremetal node set <node-uuid> --property root_device='{"vendor": "<or>_  
↳samsung <or> winsys"}'
```

(continues on next page)

(continued from previous page)

Note: If multiple hints are specified, a device must satisfy all the hints.

Appending kernel parameters to boot instances

The Bare Metal service supports passing custom kernel parameters to boot instances to fit users requirements. The way to append the kernel parameters is depending on how to boot instances.

Network boot

Currently, the Bare Metal service supports assigning unified kernel parameters to PXE booted instances by:

- Modifying the `[pxe]/kernel_append_params` configuration option, for example:

```
[pxe]
kernel_append_params = quiet splash
```

Note: The option was called `pxe_append_params` before the Xena cycle.

- Copying a template from shipped templates to another place, for example:

```
https://opendev.org/openstack/ironic/src/branch/master/ironic/drivers/
↔modules/pxe_config.template
```

Making the modifications and pointing to the custom template via the configuration options: `[pxe]/pxe_config_template` and `[pxe]/uefi_pxe_config_template`.

Local boot

For local boot instances, users can make use of configuration drive (see *Enabling the configuration drive (configdrive)*) to pass a custom script to append kernel parameters when creating an instance. This is more flexible and can vary per instance. Here is an example for grub2 with ubuntu, users can customize it to fit their use case:

```
#!/usr/bin/env python
import os

# Default grub2 config file in Ubuntu
grub_file = '/etc/default/grub'
# Add parameters here to pass to instance.
kernel_parameters = ['quiet', 'splash']
grub_cmd = 'GRUB_CMDLINE_LINUX'
old_grub_file = grub_file+'~'
```

(continues on next page)

(continued from previous page)

```

os.rename(grub_file, old_grub_file)
cmdline_existed = False
with open(grub_file, 'w') as writer, \
    open(old_grub_file, 'r') as reader:
    for line in reader:
        key = line.split('=')[0]
        if key == grub_cmd:
            #If there is already some value:
            if line.strip()[-1] == '"':
                line = line.strip()[:-1] + ' ' + ' '.join(kernel_
↳parameters) + '"'
                cmdline_existed = True
            writer.write(line)
        if not cmdline_existed:
            line = grub_cmd + '=' + '"' + ' '.join(kernel_parameters) + '"'
            writer.write(line)

os.remove(old_grub_file)
os.system('update-grub')
os.system('reboot')

```

Console

In order to change default console configuration in the Bare Metal service configuration file ([pxe] section in `/etc/ironic/ironic.conf`), include the serial port terminal and serial speed. Serial speed must be the same as the serial configuration in the BIOS settings, so that the operating system boot process can be seen in the serial console or web console. Following examples represent possible parameters for serial and web console respectively.

- Node serial console. The console parameter `console=ttyS0,115200n8` uses `ttyS0` for console output at 115200bps, 8bit, non-parity, e.g.:

```

[pxe]

# Additional append parameters for baremetal PXE boot.
kernel_append_params = nofb nomodeset vga=normal console=ttyS0,115200n8

```

- For node web console configuration is similar with the addition of `ttyX` parameter, see example:

```

[pxe]

# Additional append parameters for baremetal PXE boot.
kernel_append_params = nofb nomodeset vga=normal console=tty0,
↳console=ttyS0,115200n8

```

For detailed information on how to add consoles see the reference documents [kernel params](#) and [serial console](#). In case of local boot the Bare Metal service is not able to control kernel boot parameters. To configure console locally, follow Local boot section above.

Boot mode support

Some of the bare metal hardware types (namely, `redfish`, `ilo` and generic `ipmi`) support setting boot mode (Legacy BIOS or UEFI).

Note: Setting boot mode support in generic `ipmi` driver is coupled with setting boot device. That makes boot mode support in the `ipmi` driver incomplete.

Note: In this chapter we will distinguish *ironic node* from *bare metal node*. The difference is that *ironic node* refers to a logical node, as it is configured in ironic, while *bare metal node* indicates the hardware machine that ironic is managing.

The following rules apply in order when ironic manages node boot mode:

- If the hardware type (or bare metal node) does not implement reading current boot mode of the bare metal node, then ironic assumes that boot mode is not set on the bare metal node
- If boot mode is not set on ironic node and bare metal node boot mode is unknown (not set, cant be read etc.), ironic node boot mode is set to the value of the `[deploy]/default_boot_mode` option
- If boot mode is set on a bare metal node, but is not set on ironic node, bare metal node boot mode is set on ironic node
- If boot mode is set on ironic node, but is not set on the bare metal node, ironic node boot mode is attempted to be set on the bare metal node (failure to set boot mode on the bare metal node will not fail ironic node deployment)
- If different boot modes appear on to be set ironic node and on the bare metal node, ironic node boot mode is attempted to be set on the bare metal node (failure to set boot mode on the bare metal node will fail ironic node deployment)

Warning: If a bare metal node does not support setting boot mode, then the operator needs to make sure that boot mode configuration is consistent between ironic node and the bare metal node.

The boot modes can be configured in the Bare Metal service in the following way:

- Only one boot mode (either `uefi` or `bios`) can be configured for the node.
- If the operator wants a node to boot always in `uefi` mode or `bios` mode, then they may use `capabilities` parameter within `properties` field of an bare metal node. The operator must manually set the appropriate boot mode on the bare metal node.

To configure a node in `uefi` mode, then set `capabilities` as below:

```
openstack baremetal node set <node-uuid> --property capabilities='boot_
↳mode:uefi'
```

Nodes having `boot_mode` set to `uefi` may be requested by adding an `extra_spec` to the Compute service flavor:

```
nova flavor-key ironic-test-3 set capabilities:boot_mode="uefi"  
nova boot --flavor ironic-test-3 --image test-image instance-1
```

If `capabilities` is used in `extra_spec` as above, nova scheduler (`ComputeCapabilitiesFilter`) will match only bare metal nodes which have the `boot_mode` set appropriately in `properties/capabilities`. It will filter out rest of the nodes.

The above facility for matching in the Compute service can be used in heterogeneous environments where there is a mix of `uefi` and `bios` machines, and operator wants to provide a choice to the user regarding boot modes. If the flavor doesn't contain `boot_mode` and `boot_mode` is configured for bare metal nodes, then nova scheduler will consider all nodes and user may get either `bios` or `uefi` machine.

Some hardware support setting secure boot mode, see [UEFI secure boot mode](#) for details.

Choosing the disk label

Note: The term `disk label` is historically used in Ironic and was taken from `parted`. Apparently everyone seems to have a different word for `disk label` - these are all the same thing: `disk type`, `partition table`, `partition map` and so on

Ironic allows operators to choose which disk label they want their bare metal node to be deployed with when Ironic is responsible for partitioning the disk; therefore choosing the disk label does not apply when the image being deployed is a `whole disk image`.

There are some edge cases where someone may want to choose a specific disk label for the images being deployed, including but not limited to:

- For machines in `bios` boot mode with disks larger than 2 terabytes it's recommended to use a `gpt` disk label. That's because a capacity beyond 2 terabytes is not addressable by using the MBR partitioning type. But, although GPT claims to be backward compatible with legacy BIOS systems [that's not always the case](#).
- Operators may want to force the partitioning to be always MBR (even if the machine is deployed with boot mode `uefi`) to avoid breakage of applications and tools running on those instances.

The disk label can be configured in two ways; when Ironic is used with the Compute service or in standalone mode. The following bullet points and sections will describe both methods:

- When no disk label is provided Ironic will configure it according to the boot mode (see [Boot mode support](#)); `bios` boot mode will use `msdos` and `uefi` boot mode will use `gpt`.
- Only one disk label - either `msdos` or `gpt` - can be configured for the node.

When used with Compute service

When Ironic is used with the Compute service the disk label should be set to nodes `properties/capabilities` field and also to the flavor which will request such capability, for example:

```
baremetal node set <node-uuid> --property capabilities='disk_label:gpt'
```

As for the flavor:

```
nova flavor-key baremetal set capabilities:disk_label="gpt"
```

When used in standalone mode

When used without the Compute service, the disk label should be set directly to the nodes `instance_info` field, as below:

```
baremetal node set <node-uuid> --instance-info capabilities='{"disk_label":  
↪ "gpt"}'
```

Notifications

The Bare Metal service supports the emission of notifications, which are messages sent on a message broker (like RabbitMQ or anything else supported by the [oslo messaging library](#)) that indicate various events which occur, such as when a node changes power states. These can be consumed by an external service reading from the message bus. For example, [Searchlight](#) is an OpenStack service that uses notifications to index (and make searchable) resources from the Bare Metal service.

Notifications are disabled by default. For a complete list of available notifications and instructions for how to enable them, see the [Notifications](#).

Configuring node web console

See [Configuring Web or Serial Console](#).

2.1.11 Troubleshooting

Once all the services are running and configured properly, and a node has been enrolled with the Bare Metal service and is in the `available` provision state, the Compute service should detect the node as an available resource and expose it to the scheduler.

Note: There is a delay, and it may take up to a minute (one periodic task cycle) for the Compute service to recognize any changes in the Bare Metal services resources (both additions and deletions).

In addition to watching `nova-compute` log files, you can see the available resources by looking at the list of Compute hypervisors. The resources reported therein should match the bare metal node properties, and the Compute service flavor.

Here is an example set of commands to compare the resources in Compute service and Bare Metal service:

```

$ baremetal node list
+-----+-----+-----+-----+
↪-----+-----+
| UUID                               | Instance UUID | Power State | ↪
↪Provisioning State | Maintenance |
+-----+-----+-----+-----+
↪-----+-----+
| 86a2b1bb-8b29-4964-a817-f90031debddb | None          | power off  | ↪
↪available          | False        |
+-----+-----+-----+-----+
↪-----+-----+

$ baremetal node show 86a2b1bb-8b29-4964-a817-f90031debddb
+-----+-----+-----+-----+
↪-----+-----+
| Property                           | Value                                               | ↪
↪                                     |
+-----+-----+-----+-----+
↪-----+-----+
| instance_uuid                       | None                                               | ↪
↪                                     |
| properties                           | {u'memory_mb': u'1024', u'cpu_arch': u'x86_64', u |
↪'local_gb': u'10', | | u'cpus': u'1'} | ↪
|                                     |
| maintenance                         | False                                             | ↪
↪                                     |
| driver_info                         | { [SNIP] }                                       | ↪
↪                                     |
| extra                               | {}                                               | ↪
↪                                     |
| last_error                          | None                                             | ↪
↪                                     |
| created_at                          | 2014-11-20T23:57:03+00:00                       | ↪
↪                                     |
| target_provision_state              | None                                             | ↪
↪                                     |
| driver                              | ipmi                                             | ↪
↪                                     |
| updated_at                          | 2014-11-21T00:47:34+00:00                       | ↪
↪                                     |
| instance_info                       | {}                                               | ↪
↪                                     |
| chassis_uuid                        | 7b49bbc5-2eb7-4269-b6ea-3f1a51448a59          | ↪
↪                                     |
| provision_state                     | available                                         | ↪
↪                                     |
| reservation                         | None                                             | ↪
↪                                     |
| power_state                         | power off                                         | ↪
↪                                     |

```

(continues on next page)

(continued from previous page)

```

| console_enabled      | False
↪
| uuid                 | 86a2b1bb-8b29-4964-a817-f90031debddb
↪
+-----+-----+
↪-----+
$ nova hypervisor-list
+-----+-----+
↪+-----+-----+
| ID                               | Hypervisor hostname
↪| State | Status |
+-----+-----+
↪+-----+-----+
| 584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3 | 86a2b1bb-8b29-4964-a817-f90031debddb
↪| up   | enabled |
+-----+-----+
↪+-----+-----+
$ nova hypervisor-show 584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3
+-----+-----+
| Property                        | Value
+-----+-----+
| cpu_info                        | baremetal cpu
| current_workload                 | 0
| disk_available_least            | -
| free_disk_gb                    | 10
| free_ram_mb                     | 1024
| host_ip                         | [ SNIP ]
| hypervisor_hostname             | 86a2b1bb-8b29-4964-a817-f90031debddb
| hypervisor_type                 | ironic
| hypervisor_version              | 1
| id                               | 1
| local_gb                        | 10
| local_gb_used                   | 0
| memory_mb                       | 1024
| memory_mb_used                  | 0
| running_vms                    | 0
| service_disabled_reason         | -
| service_host                    | my-test-host
| service_id                      | 6
| state                           | up
| status                          | enabled
| vcpus                           | 1
| vcpus_used                      | 0
+-----+-----+

```

Maintenance mode

Maintenance mode may be used if you need to take a node out of the resource pool. Putting a node in maintenance mode will prevent Bare Metal service from executing periodic tasks associated with the node. This will also prevent Compute service from placing a tenant instance on the node by not exposing the node to the nova scheduler. Nodes can be placed into maintenance mode with the following command.

```
$ baremetal node maintenance set $NODE_UUID
```

A maintenance reason may be included with the optional `--reason` command line option. This is a free form text field that will be displayed in the `maintenance_reason` section of the `node show` command.

```
$ baremetal node maintenance set $UUID --reason "Need to add ram."
```

```
$ baremetal node show $UUID
```

```
+-----+-----+
| Property          | Value          |
+-----+-----+
| target_power_state | None           |
| extra              | {}             |
| last_error         | None           |
| updated_at         | 2015-04-27T15:43:58+00:00 |
| maintenance_reason | Need to add ram. |
| ...                | ...            |
| maintenance        | True           |
| ...                | ...            |
+-----+-----+
```

To remove maintenance mode and clear any `maintenance_reason` use the following command.

```
$ baremetal node maintenance unset $NODE_UUID
```

2.1.12 Next steps

Your OpenStack environment now includes the Bare Metal service.

2.1.13 Create user images for the Bare Metal service

The content has been migrated, please see *Creating instance images*.

2.2 Bare Metal Service Upgrade Guide

This document outlines various steps and notes for operators to consider when upgrading their ironic-driven clouds from previous versions of OpenStack.

The Bare Metal (ironic) service is tightly coupled with the ironic driver that is shipped with the Compute (nova) service. Some special considerations must be taken into account when upgrading your cloud.

Both offline and rolling upgrades are supported.

2.2.1 Plan your upgrade

- Rolling upgrades are available starting with the Pike release; that is, when upgrading from Ocata. This means that it is possible to do an upgrade with minimal to no downtime of the Bare Metal API.
- Upgrades are only supported between two consecutive named releases. This means that you cannot upgrade Ocata directly into Queens; you need to upgrade into Pike first.
- The [release notes](#) should always be read carefully when upgrading the Bare Metal service. Specific upgrade steps and considerations are documented there.
- The Bare Metal service should always be upgraded before the Compute service.

Note: The ironic virt driver in nova always uses a specific version of the ironic REST API. This API version may be one that was introduced in the same development cycle, so upgrading nova first may result in nova being unable to use the Bare Metal API.

- Make a backup of your database. Ironic does not support downgrading of the database. Hence, in case of upgrade failure, restoring the database from a backup is the only choice.
- Before starting your upgrade, it is best to ensure that all nodes have reached, or are in, a stable `provision_state`. Nodes in states with long running processes such as deploying or cleaning, may fail, and may require manual intervention to return them to the available hardware pool. This is most likely in cases where a timeout has occurred or a service was terminated abruptly. For a visual diagram detailing states and possible state transitions, please see *Ironic's State Machine*.

2.2.2 Offline upgrades

In an offline (or cold) upgrade, the Bare Metal service is not available during the upgrade, because all the services have to be taken down.

When upgrading the Bare Metal service, the following steps should always be taken in this order:

1. upgrade the ironic-python-agent image
2. update ironic code, without restarting services
3. run database schema migrations via `ironic-dbsync upgrade`

4. restart ironic-conductor and ironic-api services

Once the above is done, do the following:

- update any applicable configuration options to stop using any deprecated features or options, and perform any required work to transition to alternatives. All the deprecated features and options will be supported for one release cycle, so should be removed before your next upgrade is performed.
- upgrade python-ironicclient along with any other services connecting to the Bare Metal service as a client, such as nova-compute
- run the `ironic-dbsync online_data_migrations` command to make sure that data migrations are applied. The command lets you limit the impact of the data migrations with the `--max-count` option, which limits the number of migrations executed in one run. You should complete all of the migrations as soon as possible after the upgrade.

Warning: You will not be able to start an upgrade to the release after this one, until this has been completed for the current release. For example, as part of upgrading from Ocata to Pike, you need to complete Pikes data migrations. If this not done, you will not be able to upgrade to Queens it will not be possible to execute Queens database schema updates.

2.2.3 Rolling upgrades

To Reduce downtime, the services can be upgraded in a rolling fashion, meaning to upgrade one or a few services at a time to minimize impact.

Rolling upgrades are available starting with the Pike release. This feature makes it possible to upgrade between releases, such as Ocata to Pike, with minimal to no downtime of the Bare Metal API.

Requirements

To facilitate an upgrade in a rolling fashion, you need to have a highly-available deployment consisting of at least two ironic-api and two ironic-conductor services. Use of a load balancer to balance requests across the ironic-api services is recommended, as it allows for a minimal impact to end users.

Concepts

There are four aspects of the rolling upgrade process to keep in mind:

- API and RPC version pinning, and versioned object backports
- online data migrations
- graceful service shutdown
- API load balancer draining

API & RPC version pinning and versioned object backports

Through careful RPC versioning, newer services are able to talk to older services (and vice-versa). The `[DEFAULT]/pin_release_version` configuration option is used for this. It should be set (pinned) to the release version that the older services are using. The newer services will backport RPC calls and objects to their appropriate versions from the pinned release. If the `IncompatibleObjectVersion` exception occurs, it is most likely due to an incorrect or unspecified `[DEFAULT]/pin_release_version` configuration value. For example, when `[DEFAULT]/pin_release_version` is not set to the older release version, no conversion will happen during the upgrade.

For the `ironic-api` service, the API version is pinned via the same `[DEFAULT]/pin_release_version` configuration option as above. When pinned, the new `ironic-api` services will not service any API requests with Bare Metal API versions that are higher than what the old `ironic-api` services support. HTTP status code 406 is returned for such requests. This prevents new features (available in new API versions) from being used until after the upgrade has been completed.

Online data migrations

To make database schema migrations less painful to execute, we have implemented process changes to facilitate upgrades.

- All data migrations are banned from schema migration scripts.
- Schema migration scripts only update the database schema.
- Data migrations must be done at the end of the rolling upgrade process, after the schema migration and after the services have been upgraded to the latest release.

All data migrations are performed using the `ironic-dbsync online_data_migrations` command. It can be run as a background process so that it does not interrupt running services; however it must be run to completion for a cold upgrade if the intent is to make use of new features immediately.

(You would also execute the same command with services turned off if you are doing a cold upgrade).

This data migration must be completed. If not, you will not be able to upgrade to future releases. For example, if you had upgraded from Ocata to Pike but did not do the data migrations, you will not be able to upgrade from Pike to Queens. (More precisely, you will not be able to apply Queens schema migrations.)

Graceful conductor service shutdown

The `ironic-conductor` service is a Python process listening for messages on a message queue. When the operator sends the `SIGTERM` signal to the process, the service stops consuming messages from the queue, so that no additional work is picked up. It completes any outstanding work and then terminates. During this process, messages can be left on the queue and will be processed after the Python process starts back up. This gives us a way to shutdown a service using older code, and start up a service using newer code with minimal impact.

Note: This was tested with RabbitMQ messaging backend and may vary with other backends.

Nodes that are being acted upon by an `ironic-conductor` process, which are not in a stable state, may encounter failures. Node failures that occur during an upgrade are likely due to timeouts, resulting from

delays involving messages being processed and acted upon by a conductor during long running, multi-step processes such as deployment or cleaning.

API load balancer draining

If you are using a load balancer for the ironic-api services, we recommend that you redirect requests to the new API services and drain off of the ironic-api services that have not yet been upgraded.

Rolling upgrade process

Before maintenance window

- Upgrade the ironic-python-agent image
- Using the new release (ironic code), execute the required database schema updates by running the database upgrade command: `ironic-dbsync upgrade`. These schema change operations should have minimal or no effect on performance, and should not cause any operations to fail (but please check the release notes). You can:
 - install the new release on an existing system
 - install the new release in a new virtualenv or a container

At this point, new columns and tables may exist in the database. These database schema changes are done in a way that both the old and new (N and N+1) releases can perform operations against the same schema.

Note: Ironic bases its API, RPC and object storage format versions on the `[DEFAULT]/pin_release_version` configuration option. It is advisable to automate the deployment of changes in configuration files to make the process less error prone and repeatable.

During maintenance window

1. All ironic-conductor services should be upgraded first. Ensure that at least one ironic-conductor service is running at all times. For every ironic-conductor, either one by one or a few at a time:
 - shut down the service. Messages from the ironic-api services to the conductors are load-balanced by the message queue and a hash-ring, so the only thing you need to worry about is to shut the service down gracefully (using `SIGTERM` signal) to make sure it will finish all the requests being processed before shutting down.
 - upgrade the installed version of ironic and dependencies
 - set the `[DEFAULT]/pin_release_version` configuration option value to the version you are upgrading from (that is, the old version). Based on this setting, the new ironic-conductor services will downgrade any RPC communication and data objects to conform to the old service. For example, if you are upgrading from Ocata to Pike, set this value to `ocata`.
 - start the service

2. The next service to upgrade is `ironic-api`. Ensure that at least one `ironic-api` service is running at all times. You may want to start another temporary instance of the older `ironic-api` to handle the load while you are upgrading the original `ironic-api` services. For every `ironic-api` service, either one by one or a few at a time:
 - in HA deployment you are typically running them behind a load balancer (for example HAProxy), so you need to take the service instance out of the balancer
 - shut it down
 - upgrade the installed version of `ironic` and dependencies
 - set the `[DEFAULT]/pin_release_version` configuration option value to the version you are upgrading from (that is, the old version). Based on this setting, the new `ironic-api` services will downgrade any RPC communication and data objects to conform to the old service. In addition, the new services will return HTTP status code 406 for any requests with newer API versions that the old services did not support. This prevents new features (available in new API versions) from being used until after the upgrade has been completed. For example, if you are upgrading from Ocata to Pike, set this value to `ocata`.
 - restart the service
 - add it back into the load balancer

After upgrading all the `ironic-api` services, the Bare Metal service is running in the new version but with downgraded RPC communication and database object storage formats. New features (in new API versions) are not supported, because they could fail when objects are in the downgraded object formats and some internal RPC API functions may still not be available.

3. For all the `ironic-conductor` services, one at a time:
 - remove the `[DEFAULT]/pin_release_version` configuration option setting
 - restart the `ironic-conductor` service
4. For all the `ironic-api` services, one at a time:
 - remove the `[DEFAULT]/pin_release_version` configuration option setting
 - restart the `ironic-api` service

After maintenance window

Now that all the services are upgraded, the system is able to use the latest version of the RPC protocol and able to access all the features of the new release.

- Update any applicable configuration options to stop using any deprecated features or options, and perform any required work to transition to alternatives. All the deprecated features and options will be supported for one release cycle, so should be removed before your next upgrade is performed.
- Upgrade `python-ironicclient` along with other services connecting to the Bare Metal service as a client, such as `nova-compute`.

Warning: A `nova-compute` instance tries to attach VIFs to all active instances on start up. Make sure that for all active nodes there is at least one running `ironic-conductor` process to

manage them. Otherwise the instances will be moved to the ERROR state on the nova-compute start up.

- Run the `ironic-dbsync online_data_migrations` command to make sure that data migrations are applied. The command lets you limit the impact of the data migrations with the `--max-count` option, which limits the number of migrations executed in one run. You should complete all of the migrations as soon as possible after the upgrade.

Warning: Note that you will not be able to start an upgrade to the next release after this one, until this has been completed for the current release. For example, as part of upgrading from Ocata to Pike, you need to complete Pikes data migrations. If this not done, you will not be able to upgrade to Queens it will not be possible to execute Queens database schema updates.

Upgrading to Hardware Types

Starting with the Rocky release, the Bare Metal service does not support *classic drivers* any more. If you still use *classic drivers*, please upgrade to *hardware types* immediately. Please see [Enabling drivers and hardware types](#) for details on *hardware types* and *hardware interfaces*.

Planning the upgrade

It is necessary to figure out which hardware types and hardware interfaces correspond to which classic drivers used in your deployment. The following table lists the classic drivers with their corresponding hardware types and the boot, deploy, inspect, management, and power hardware interfaces:

Classic Driver	Hardware Type	Boot	De- ploy	Inspect	Manage- ment	Power
agent_ilo	ilo	ilo-virtual- media	direct	ilo	ilo	ilo
agent_ipmitool	ipmi	pxe	direct	inspec- tor	ipmitool	ipmi- tool
agent_ipmitool_socat	ipmi	pxe	direct	inspec- tor	ipmitool	ipmi- tool
agent_irmc	irmc	irmc-virtual- media	direct	irmc	irmc	irmc
iscsi_ilo	ilo	ilo-virtual- media	iscsi	ilo	ilo	ilo
iscsi_irmc	irmc	irmc-virtual- media	iscsi	irmc	irmc	irmc
pxe_drac	idrac	pxe	iscsi	idrac	idrac	idrac
pxe_drac_inspector	idrac	pxe	iscsi	inspec- tor	idrac	idrac
pxe_ilo	ilo	ilo-pxe	iscsi	ilo	ilo	ilo
pxe_ipmitool	ipmi	pxe	iscsi	inspec- tor	ipmitool	ipmi- tool
pxe_ipmitool_socat	ipmi	pxe	iscsi	inspec- tor	ipmitool	ipmi- tool
pxe_irmc	irmc	irmc-pxe	iscsi	irmc	irmc	irmc
pxe_snmp	snmp	pxe	iscsi	no- inspect	fake	snmp

Note: The inspector *inspect* interface was only used if explicitly enabled in the configuration. Otherwise, *no-inspect* was used.

Note: `pxe_ipmitool_socat` and `agent_ipmitool_socat` use `ipmitool-socat console` interface (the default for the ipmi hardware type), while `pxe_ipmitool` and `agent_ipmitool` use `ipmitool-shellinbox`. See [Console](#) for details.

For out-of-tree drivers you may need to reach out to their maintainers or figure out the appropriate interfaces by researching the source code.

Configuration

You will need to enable hardware types and interfaces that correspond to your currently enabled classic drivers. For example, if you have the following configuration in your `ironic.conf`:

```
[DEFAULT]
enabled_drivers = pxe_ipmitool,agent_ipmitool
```

You will have to add this configuration as well:

```
[DEFAULT]
enabled_hardware_types = ipmi
enabled_boot_interfaces = pxe
enabled_deploy_interfaces = iscsi,direct
enabled_management_interfaces = ipmitool
enabled_power_interfaces = ipmitool
```

Note: For every interface type there is an option `default_<INTERFACE>_interface`, where `<INTERFACE>` is the interface type name. For example, one can make all nodes use the `direct` deploy method by default by setting:

```
[DEFAULT]
default_deploy_interface = direct
```

Migrating nodes

After the required items are enabled in the configuration, each nodes `driver` field has to be updated to a new value. You may need to also set new values for some or all interfaces:

```
export OS_BAREMETAL_API_VERSION=1.31

for uuid in $(baremetal node list --driver pxe_ipmitool -f value -c UUID); do
    baremetal node set <node> --driver ipmi --deploy-interface iscsi
done

for uuid in $(baremetal node list --driver agent_ipmitool -f value -c UUID); do
    baremetal node set <node> --driver ipmi --deploy-interface direct
done
```

See *Enrollment* for more details on setting hardware types and interfaces.

Warning: It is not recommended to change the interfaces for active nodes. If absolutely needed, the nodes have to be put in the maintenance mode first:

```
baremetal node maintenance set $UUID \
    --reason "Changing driver and/or hardware interfaces"
# do the update, validate its correctness
baremetal node maintenance unset $UUID
```

Other interfaces

Care has to be taken to migrate from classic drivers using non-default interfaces. This chapter covers a few of the most commonly used.

Ironic Inspector

Some classic drivers, notably `pxe_ipmitool`, `agent_ipmitool` and `pxe_drac_inspector`, use `ironic-inspector` for their `inspect` interface.

The same functionality is available for all hardware types, but the appropriate `inspect` interface has to be enabled in the Bare Metal service configuration file, for example:

```
[DEFAULT]
enabled_inspect_interfaces = inspector,no-inspect
```

See *Enabling drivers and hardware types* for more details.

Note: The configuration option `[inspector]enabled` does not affect hardware types.

Then you can tell your nodes to use this interface, for example:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --inspect-interface inspector
done
```

Note: A node configured with the IPMI hardware type, will use the inspector inspection implementation automatically if it is enabled. This is not the case for the most of the vendor drivers.

Console

Several classic drivers, notably `pxe_ipmitool_socat` and `agent_ipmitool_socat`, use `socat`-based serial console implementation.

For the `ipmi` hardware type it is used by default, if enabled in the configuration file:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-socat,no-console
```

If you want to use the `shellinabox` implementation instead, it has to be enabled as well:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-shellinabox,no-console
```

Then you need to update some or all nodes to use it explicitly. For example, to update all nodes use:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --console-interface ipmitool-shellinbox
done
```

RAID

Many classic drivers, including `pxe_ipmitool` and `agent_ipmitool` use the IPA-based in-band RAID implementation by default.

For the hardware types it is not used by default. To use it, you need to enable it in the configuration first:

```
[DEFAULT]
enabled_raid_interfaces = agent,no-raid
```

Then you can update those nodes that support in-band RAID to use the agent RAID interface. For example, to update all nodes use:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --raid-interface agent
done
```

Note: The ability of a node to use the agent RAID interface depends on the ramdisk (more specifically, a [hardware manager](#) used in it), not on the driver.

Network and storage

The network and storage interfaces have always been dynamic, and thus do not require any special treatment during upgrade.

Vendor

Classic drivers are allowed to use the `VendorMixin` functionality to combine and expose several node or driver vendor passthru methods from different vendor interface implementations in one driver.

This is no longer possible with hardware types.

With hardware types, a vendor interface can only have a single active implementation from the list of vendor interfaces supported by a given hardware type.

Ironic no longer has in-tree drivers (both classic and hardware types) that rely on this `VendorMixin` functionality support. However if you are using an out-of-tree classic driver that depends on it, you'll need to do the following in order to use vendor passthru methods from different vendor passthru implementations:

1. While creating a new hardware type to replace your classic driver, specify all vendor interface implementations your classic driver was using to build its `VendorMixin` as supported vendor interfaces (property `supported_vendor_interfaces` of the Python class that defines your hardware type).

2. Ensure all required vendor interfaces are enabled in the ironic configuration file under the [DEFAULT]enabled_vendor_interfaces option. You should also consider setting the [DEFAULT]default_vendor_interface option to specify the vendor interface for nodes that do not have one set explicitly.
3. Before invoking a specific vendor passthru method, make sure that the nodes vendor interface is set to the interface with the desired vendor passthru method. For example, if you want to invoke the vendor passthru method vendor_method_foo() from vendor_foo vendor interface:

```
# set the vendor interface to 'vendor_foo`
baremetal node set <node> --vendor-interface vendor_foo

# invoke the vendor passthru method
baremetal node passthru call <node> vendor_method_foo
```


3.1 Bare Metal Service User Guide

Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines. It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance) and Object (swift) services.

When the Bare Metal service is appropriately configured with the Compute and Network services, it is possible to provision both virtual and physical machines through the Compute services API. However, the set of instance actions is limited, arising from the different characteristics of physical servers and switch hardware. For example, live migration can not be performed on a bare metal instance.

The community maintains reference drivers that leverage open-source technologies (eg. PXE and IPMI) to cover a wide range of hardware. Ironics pluggable driver architecture also allows hardware vendors to write and contribute drivers that may improve performance or add functionality not provided by the community drivers.

3.1.1 Understanding Bare Metal service

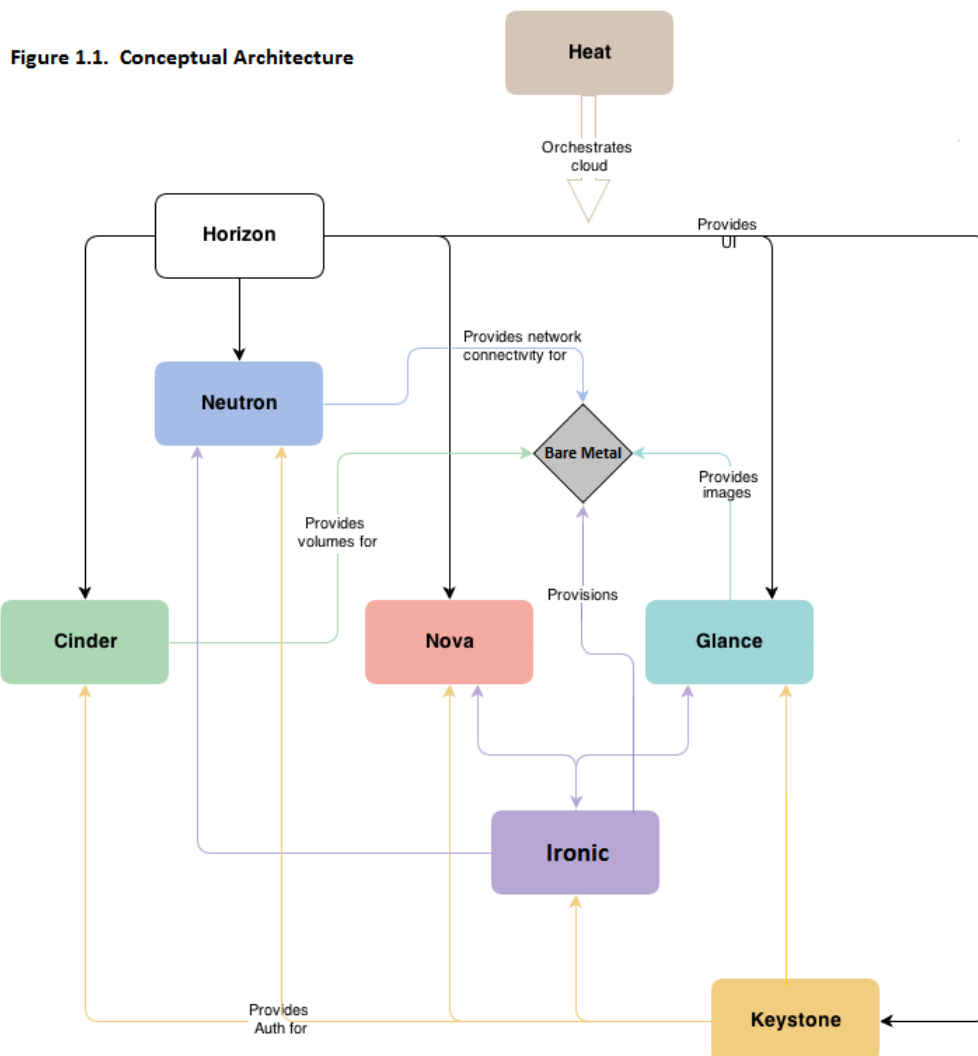
Why Provision Bare Metal

Here are a few use-cases for bare metal (physical server) provisioning in cloud; there are doubtless many more interesting ones:

- High-performance computing clusters
- Computing tasks that require access to hardware devices which cant be virtualized
- Database hosting (some databases run poorly in a hypervisor)
- Single tenant, dedicated hardware for performance, security, dependability and other regulatory requirements
- Or, rapidly deploying a cloud infrastructure

Conceptual Architecture

The following diagram shows the relationships and how all services come into play during the provisioning of a physical server. (Note that Ceilometer and Swift can be used with Ironic, but are missing from this diagram.)



Key Technologies for Bare Metal Hosting

Preboot Execution Environment (PXE)

PXE is part of the Wired for Management (WfM) specification developed by Intel and Microsoft. The PXE enables systems BIOS and network interface card (NIC) to bootstrap a computer from the network in place of a disk. Bootstrapping is the process by which a system loads the OS into local memory so that it can be executed by the processor. This capability of allowing a system to boot over a network simplifies server deployment and server management for administrators.

Dynamic Host Configuration Protocol (DHCP)

DHCP is a standardized networking protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. Using PXE, the BIOS uses DHCP to obtain an IP address for the network interface and to locate the server that stores the network bootstrap program (NBP).

Network Bootstrap Program (NBP)

NBP is equivalent to GRUB (GRand Unified Bootloader) or LILO (LInux LOader) - loaders which are traditionally used in local booting. Like the boot program in a hard drive environment, the NBP is responsible for loading the OS kernel into memory so that the OS can be bootstrapped over a network.

Trivial File Transfer Protocol (TFTP)

TFTP is a simple file transfer protocol that is generally used for automated transfer of configuration or boot files between machines in a local environment. In a PXE environment, TFTP is used to download NBP over the network using information from the DHCP server.

Intelligent Platform Management Interface (IPMI)

IPMI is a standardized computer system interface used by system administrators for out-of-band management of computer systems and monitoring of their operation. It is a method to manage systems that may be unresponsive or powered off by using only a network connection to the hardware rather than to an operating system.

Understanding Bare Metal Deployment

What happens when a boot instance request comes in? The below diagram walks through the steps involved during the provisioning of a bare metal instance.

These pre-requisites must be met before the deployment process:

- Dependent packages to be configured on the Bare Metal service node(s) where ironic-conductor is running like tftp-server, ipmi, syslinux etc for bare metal provisioning.
- Nova must be configured to make use of the bare metal service endpoint and compute driver should be configured to use ironic driver on the Nova compute node(s).
- Flavors to be created for the available hardware. Nova must know the flavor to boot from.
- Images to be made available in Glance. Listed below are some image types required for successful bare metal deployment:
 - bm-deploy-kernel
 - bm-deploy-ramdisk
 - user-image
 - user-image-vmlinuz
 - user-image-initrd

- Hardware to be enrolled via the bare metal API service.

Deploy Process

This describes a typical bare metal node deployment within OpenStack using PXE to boot the ramdisk. Depending on the ironic driver interfaces used, some of the steps might be marginally different, however the majority of them will remain the same.

1. A boot instance request comes in via the Nova API, through the message queue to the Nova scheduler.
2. Nova scheduler applies filters and finds the eligible hypervisor. The nova scheduler also uses the flavors `extra_specs`, such as `cpu_arch`, to match the target physical node.
3. Nova compute manager claims the resources of the selected hypervisor.
4. Nova compute manager creates (unbound) tenant virtual interfaces (VIFs) in the Networking service according to the network interfaces requested in the nova boot request. A caveat here is, the MACs of the ports are going to be randomly generated, and will be updated when the VIF is attached to some node to correspond to the node network interface cards (or bonds) MAC.
5. A spawn task is created by the nova compute which contains all the information such as which image to boot from etc. It invokes the `driver.spawn` from the virt layer of Nova compute. During the spawn process, the virt driver does the following:
 1. Updates the target ironic node with the information about deploy image, instance UUID, requested capabilities and various flavor properties.
 2. Validates nodes power and deploy interfaces, by calling the ironic API.
 3. Attaches the previously created VIFs to the node. Each neutron port can be attached to any ironic port or port group, with port groups having higher priority than ports. On ironic side, this work is done by the network interface. Attachment here means saving the VIF identifier into ironic port or port group and updating VIF MAC to match the ports or port groups MAC, as described in bullet point 4.
 4. Generates config drive, if requested.
6. Novas ironic virt driver issues a deploy request via the Ironic API to the Ironic conductor servicing the bare metal node.
7. Virtual interfaces are plugged in and Neutron API updates DHCP port to set PXE/TFTP options. In case of using `neutron` network interface, ironic creates separate provisioning ports in the Networking service, while in case of `flat` network interface, the ports created by nova are used both for provisioning and for deployed instance networking.
8. The ironic nodes boot interface prepares (i)PXE configuration and caches deploy kernel and ramdisk.
9. The ironic nodes management interface issues commands to enable network boot of a node.
10. The ironic nodes deploy interface caches the instance image, kernel and ramdisk if needed (it is needed in case of netboot for example).
11. The ironic nodes power interface instructs the node to power on.
12. The node boots the deploy ramdisk.

13. Depending on the exact driver used, the deploy ramdisk downloads the image from a URL (*Direct deploy*) or the conductor uses SSH to execute commands (*Ansible deploy*). The URL can be generated by Swift API-compatible object stores, for example Swift itself or RadosGW, or provided by a user.

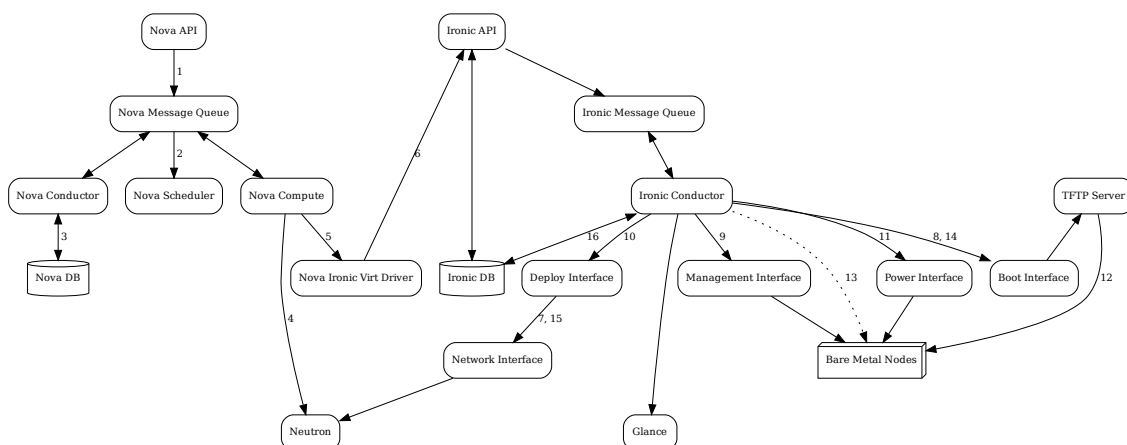
The image deployment is done.

14. The nodes boot interface switches pxe config to refer to instance images (or, in case of local boot, sets boot device to disk), and asks the ramdisk agent to soft power off the node. If the soft power off by the ramdisk agent fails, the bare metal node is powered off via IPMI/BMC call.
15. The deploy interface triggers the network interface to remove provisioning ports if they were created, and binds the tenant ports to the node if not already bound. Then the node is powered on.

Note: There are 2 power cycles during bare metal deployment; the first time the node is powered-on when ramdisk is booted, the second time after the image is deployed.

16. The bare metal nodes provisioning state is updated to active.

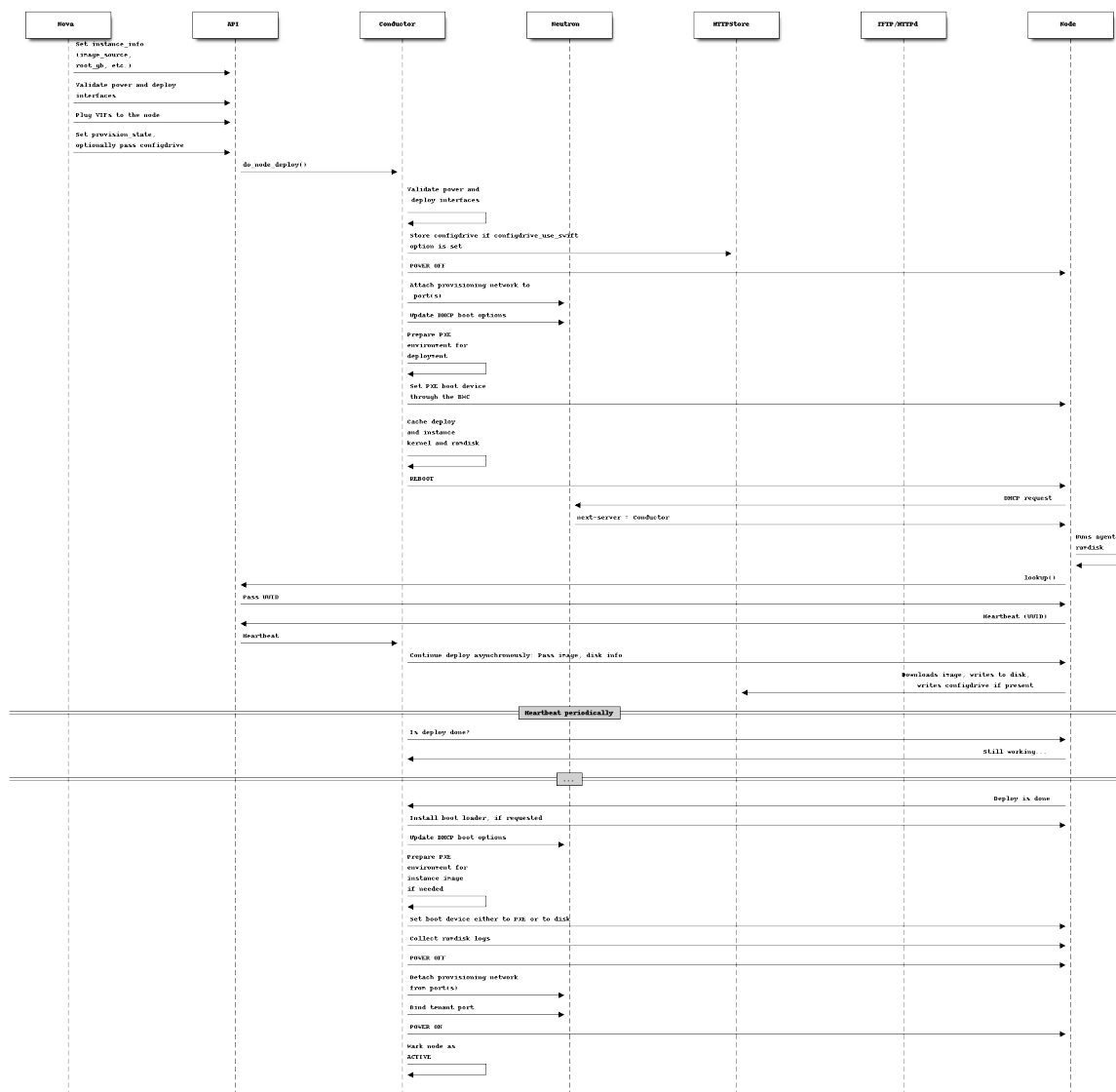
Below is the diagram that describes the above process.



The following two examples describe what ironic is doing in more detail, leaving out the actions performed by nova and some of the more advanced options.

Example: PXE Boot and Direct Deploy Process

This process is how *Direct deploy* works.



(From a talk and slides)

3.1.2 Creating instance images

Bare Metal provisioning requires two sets of images: the deploy images and the user images. The *deploy images* are used by the Bare Metal service to prepare the bare metal server for actual OS deployment. Whereas the user images are installed on the bare metal server to be used by the end user. There are two types of user images:

partition images contain only the contents of the root partition. Additionally, two more images are used together with them: an image with a kernel and with an initramfs.

Warning: To use partition images with local boot, Grub2 must be installed on them.

whole disk images contain a complete partition table with one or more partitions.

Warning: The kernel/initramfs pair must not be used with whole disk images, otherwise theyll be mistaken for partition images.

Many distributions publish their own cloud images. These are usually whole disk images that are built for legacy boot mode (not UEFI), with Ubuntu being an exception (they publish images that work in both modes).

disk-image-builder

The `disk-image-builder` can be used to create user images required for deployment and the actual OS which the user is going to run.

- Install `diskimage-builder` package (use `virtualenv`, if you dont want to install anything globally):

```
# pip install diskimage-builder
```

- Build the image your users will run (Ubuntu image has been taken as an example):

- Partition images

```
$ disk-image-create ubuntu baremetal dhcp-all-interfaces grub2 -o my-
↪image
```

- Whole disk images

```
$ disk-image-create ubuntu vm dhcp-all-interfaces -o my-image
```

with an EFI partition:

```
$ disk-image-create ubuntu vm block-device-efi dhcp-all-interfaces -
↪o my-image
```

The partition image command creates `my-image.qcow2`, `my-image.vmlinuz` and `my-image.initrd` files. The `grub2` element in the partition image creation command is only needed if local boot will be used to deploy `my-image.qcow2`, otherwise the images `my-image.vmlinuz` and `my-image.initrd` will be used for PXE booting after deploying the bare metal with `my-image.qcow2`. For whole disk images only the main image is used.

If you want to use Fedora image, replace `ubuntu` with `fedora` in the chosen command.

Virtual machine

Virtual machine software can also be used to build user images. There are different software options available, `qemu-kvm` is usually a good choice on linux platform, it supports emulating many devices and even building images for architectures other than the host machine by software emulation. `VirtualBox` is another good choice for non-linux host.

The procedure varies depending on the software used, but the steps for building an image are similar, the user creates a virtual machine, and installs the target system just like what is done for a real hardware. The system can be highly customized like partition layout, drivers or software shipped, etc.

Usually `libvirt` and its management tools are used to make interaction with `qemu-kvm` easier, for example, to create a virtual machine with `virt-install`:

```
$ virt-install --name centos8 --ram 4096 --vcpus=2 -f centos8.qcow2 \  
> --cdrom CentOS-8-x86_64-1905-dvd1.iso
```

Graphic frontend like `virt-manager` can also be utilized.

The disk file can be used as user image after the system is set up and powered off. The path of the disk file varies depending on the software used, usually its stored in a user-selected part of the local file system. For `qemu-kvm` or GUI frontend building upon it, its typically stored at `/var/lib/libvirt/images`.

3.1.3 Deploying with Bare Metal service

This guide explains how to use Ironic to deploy nodes without any front-end service, such as `OpenStack Compute (nova)` or `Metal3`.

Note: To simplify this task you can use the `metalsmith` tool which provides a convenient CLI for the most common cases.

Allocations

Allocation is a way to find and reserve a node suitable for deployment. When an allocation is created, the list of available nodes is searched for a node with the given *resource class* and *traits*, similarly to how it is done in *OpenStack Compute flavors*. Only the resource class is mandatory, for example:

```
$ baremetal allocation create --resource-class baremetal --wait  
+-----+  
| Field          | Value                                |  
+-----+  
| candidate_nodes | []                                    |  
| created_at      | 2019-04-03T12:18:26+00:00           |  
| extra           | {}                                    |  
| last_error      | None                                  |  
| name            | None                                  |  
| node_uuid       | 5d946337-b1d9-4b06-8eda-4fb77e994a0d |  
| resource_class  | baremetal                             |  
| state           | active                                |  
| traits          | []                                    |
```

(continues on next page)

(continued from previous page)

updated_at	2019-04-03T12:18:26+00:00	
uuid	e84f5d60-84f1-4701-a635-10ff90e2f3b0	
+-----+	+-----+	+-----+

Note: The allocation processing is fast but nonetheless asynchronous. Use the `--wait` argument to wait for the results.

If an allocation is successful, it sets the nodes `instance_uuid` to the allocation UUID. The nodes UUID can be retrieved from the allocations `node_uuid` field.

An allocation is automatically deleted when the associated node is unprovisioned. If you don't provision the node, you're responsible for deleting the allocation.

See the [allocation API reference](#) for more information on how to use allocations.

Populating instance information

The nodes `instance_info` field is a JSON object that contains all information required for deploying an instance on bare metal. It has to be populated before deployment and is automatically cleared on tear down.

Image information

You need to specify image information in the nodes `instance_info` (see *Creating instance images*):

- `image_source` - URL of the whole disk or root partition image, mandatory. The following schemes are supported: `http://`, `https://` and `file://`. Files have to be accessible by the conductor. If the scheme is missing, an Image Service (glance) image UUID is assumed.
- `root_gb` - size of the root partition, required for partition images.

Note: Older versions of the Bare Metal service used to require a positive integer for `root_gb` even for whole-disk images. You may want to set it for compatibility.

- `image_checksum` - MD5 checksum of the image specified by `image_source`, only required for `http://` images when using *Direct deploy*.

Other checksum algorithms are supported via the `image_os_hash_algo` and `image_os_hash_value` fields. They may be used instead of the `image_checksum` field.

Warning: If your operating system is running in FIPS 140-2 mode, MD5 will not be available, and you **must** use SHA256 or another modern algorithm.

Starting with the Stein release of `ironic-python-agent` can also be a URL to a checksums file, e.g. one generated with:

```
$ cd /path/to/http/root
$ md5sum *.img > checksums
```

- `kernel`, `ramdisk` - HTTP(s) or file URLs of the kernel and initramfs of the target OS. Must be added **only** for partition images. Supports the same schemes as `image_source`.

An example for a partition image:

```
baremetal node set $NODE_UUID \
--instance-info image_source=http://image.server/my-image.qcow2 \
--instance-info image_checksum=1f9c0e1bad977a954ba40928c1e11f33 \
--instance-info kernel=http://image.server/my-image.kernel \
--instance-info ramdisk=http://image.server/my-image.initramfs \
--instance-info root_gb=10
```

With a SHA256 hash:

```
baremetal node set $NODE_UUID \
--instance-info image_source=http://image.server/my-image.qcow2 \
--instance-info image_os_hash_algo=sha256 \
--instance-info image_os_hash_
↪value=a64dd95e0c48e61ed741ff026d8c89ca38a51f3799955097c5123b1705ef13d4 \
--instance-info kernel=http://image.server/my-image.kernel \
--instance-info ramdisk=http://image.server/my-image.initramfs \
--instance-info root_gb=10
```

With a whole disk image and a checksum URL:

```
baremetal node set $NODE_UUID \
--instance-info image_source=http://image.server/my-image.qcow2 \
--instance-info image_checksum=http://image.server/my-image.qcow2.CHECKSUM
```

Note: Certain hardware types and interfaces may require additional or different fields to be provided. See specific guides under *Drivers, Hardware Types and Hardware Interfaces*.

When using low RAM nodes with `http://` images that are not in the RAW format, you may want them cached locally, converted to raw and served from the conductor's HTTP server:

```
baremetal node set $NODE_UUID --instance-info image_download_source=local
```

For software RAID with whole-disk images, the root UUID of the root partition has to be provided so that the bootloader can be correctly installed:

```
baremetal node set $NODE_UUID --instance-info image_rootfs_uuid=<uuid>
```

Capabilities

- *Boot mode* can be specified per instance:

```
baremetal node set $NODE_UUID \
  --instance-info capabilities='{"boot_mode": "uefi"}'
```

Otherwise, the `boot_mode` capability from the nodes `properties` will be used.

Warning: The two settings must not contradict each other.

Note: This capability was introduced in the Wallaby release series, previously ironic used a separate `instance_info/deploy_boot_mode` field instead.

- To override the *boot option* used for this instance, set the `boot_option` capability:

```
baremetal node set $NODE_UUID \
  --instance-info capabilities='{"boot_option": "local"}'
```

- Starting with the Ussuri release, you can set *root device hints* per instance:

```
baremetal node set $NODE_UUID \
  --instance-info root_device='{"wwn": "0x4000cca77fc4dba1"}'
```

This setting overrides any previous setting in `properties` and will be removed on undeployment.

Overriding a hardware interface

Non-admins with temporary access to a node, may wish to specify different node interfaces. However, allowing them to set these interface values directly on the node is problematic, as there is no automated way to ensure that the original interface values are restored.

In order to temporarily override a hardware interface, simply set the appropriate value in `instance_info`. For example, if you'd like to override a nodes storage interface, run the following:

```
baremetal node set $NODE_UUID --instance-info storage_interface=cinder
```

`instance_info` values persist until after a node is cleaned.

Note: This feature is available starting with the Wallaby release.

Attaching virtual interfaces

If using the OpenStack Networking service (neutron), you can attach its ports to a node before deployment as VIFs:

```
baremetal node vif attach $NODE_UUID $PORT_UUID
```

Warning: These are **neutron** ports, not **ironic** ports!

VIFs are automatically detached on deprovisioning.

Deployment

1. Validate that all parameters are correct:

```
$ baremetal node validate $NODE_UUID
+-----+-----+-----+
↪-----+
| Interface | Result | Reason |
↪          |
+-----+-----+-----+
↪-----+
| boot      | True   |         |
↪          |
| console   | False  | Missing 'ipmi_terminal_port' parameter in node's
↪driver_info. |
| deploy    | True   |         |
↪          |
| inspect   | True   |         |
↪          |
| management | True   |         |
↪          |
| network   | True   |         |
↪          |
| power     | True   |         |
↪          |
| raid      | True   |         |
↪          |
| storage   | True   |         |
↪          |
+-----+-----+-----+
↪-----+
```

2. Now you can start the deployment, run:

```
baremetal node deploy $NODE_UUID
```

3. Starting with the Wallaby release you can also request custom deploy steps, see *Requesting steps* for details.

Deploying with a config drive

The configuration drive is a small image used to store instance-specific metadata and is present to the instance as a disk partition labeled `config-2`. See *Enabling the configuration drive (configdrive)* for a detailed explanation.

A configuration drive can be provided either as a whole ISO 9660 image or as JSON input for building an image. A first-boot service, such as `cloud-init`, must be running on the instance image for the configuration to be applied.

Building a config drive on the client side

For the format of the configuration drive, Bare Metal service expects a gzipped and base64 encoded ISO 9660 file with a `config-2` label. The `baremetal client` can generate a configuration drive in the *expected format*. Pass a directory path containing the files that will be injected into it via the `--config-drive` parameter of the `baremetal node deploy` command, for example:

```
baremetal node deploy $NODE_UUID --config-drive /dir/configdrive_files
```

Note: A configuration drive could also be a data block with a VFAT filesystem on it instead of ISO 9660. But its unlikely that it would be needed since ISO 9660 is widely supported across operating systems.

Building a config drive on the conductor side

Starting with the Stein release and *ironicclient 2.7.0*, you can request building a configdrive on the server side by providing a JSON with keys `meta_data`, `user_data` and `network_data` (all optional), e.g.:

```
baremetal node deploy $node_identifier \  
  --config-drive '{"meta_data": {"hostname": "server1.cluster"}}'
```

Note: When this feature is used, host name defaults to the nodes name or UUID.

SSH public keys can be provided as a mapping:

```
baremetal node deploy $NODE_UUID \  
  --config-drive '{"meta_data": {"public_keys": {"0": "ssh key contents"}}}'
```

If using `cloud-init`, its configuration can be supplied as `user_data`, e.g.:

```
baremetal node deploy $NODE_UUID \  
  --config-drive '{"user_data": "#cloud-config\n{\n  \"users\": [\n    {\n      \"name\": ...  
    }\n  ]\n}"}'
```

Warning: User data is a string, not a JSON! Also note that a prefix, such as `#cloud-config`, is required, see *user data format*.

Some first-boot services support network configuration in the [OpenStack network data format](#). It can be provided in the `network_data` field of the configuration drive.

Ramdisk booting

Advanced operators, specifically ones working with ephemeral workloads, may find it more useful to explicitly treat a node as one that would always boot from a Ramdisk. See *[Booting a Ramdisk or an ISO](#)* for details.

ADMINISTRATOR GUIDE

4.1 Drivers, Hardware Types and Hardware Interfaces

4.1.1 Generic Interfaces

Boot interfaces

The boot interface manages booting of both the deploy ramdisk and the user instances on the bare metal node.

The *PXE boot* interface is generic and works with all hardware that supports booting from network. Alternatively, several vendors provide *virtual media* implementations of the boot interface. They work by pushing an ISO image to the nodes [management controller](#), and do not require either PXE or iPXE. Check your driver documentation at *Drivers, Hardware Types and Hardware Interfaces* for details.

PXE boot

The `pxe` boot interface uses [PXE](#) or [iPXE](#) to deliver the target kernel/ramdisk pair. PXE uses relatively slow and unreliable TFTP protocol for transfer, while iPXE uses HTTP. The downside of iPXE is that its less common, and usually requires bootstrapping using PXE first.

The `pxe` boot interface works by preparing a PXE/iPXE environment for a node on the file system, then instructing the DHCP provider (for example, the Networking service) to boot the node from it. See [ref:direct-deploy-example](#) for a better understanding of the whole deployment process.

Note: Both PXE and iPXE are configured differently, when UEFI boot is used instead of conventional BIOS boot. This is particularly important for CPU architectures that do not have BIOS support at all.

The `pxe` boot interface is used by default for many hardware types, including `ipmi`. Some hardware types, notably `ilo` and `irmc` have their specific implementations of the PXE boot interface.

Additional configuration is required for this boot interface - see [Configuring PXE and iPXE](#) for details.

Common options

Enable persistent boot device for deploy/clean operation

For (i)PXE booting, Ironic uses non-persistent boot order changes for clean/deploy by default. For some drivers, persistent changes are far more costly than non-persistent ones, so this approach can bring a performance benefit.

In order to control this behavior, however, Ironic provides the `force_persistent_boot_device` flag in the nodes `driver_info`. It allows the values `Default` (make all changes but the last one upon deployment non-persistent), `Always` (make all changes persistent), and `Never` (make all boot order changes non-persistent). For example in order to have only persistent changes one would need to set something like:

```
$ openstack baremetal node set --driver-info force_persistent_boot_device=  
↔ 'Always' <node>
```

Note: It is recommended to check if the nodes state has not changed as there is no way of locking the node between these commands.

Note: The values `True/False` for the option `force_persistent_boot_device` in the nodes driver info for the (i)PXE drivers are deprecated and support for them may be removed in a future release. The former default value `False` is replaced by the new value `Default`, the value `True` is replaced by `Always`.

Deploy Interfaces

A *deploy* interface plays a critical role in the provisioning process. It orchestrates the whole deployment and defines how the image gets transferred to the target disk.

Direct deploy

With `direct` deploy interface, the deploy ramdisk fetches the image from an HTTP location. It can be an object storage (swift or RadosGW) temporary URL or a user-provided HTTP URL. The deploy ramdisk then copies the image to the target disk. See *direct deploy diagram* for a detailed explanation of how this deploy interface works.

You can specify this deploy interface when creating or updating a node:

```
baremetal node create --driver ipmi --deploy-interface direct  
baremetal node set <NODE> --deploy-interface direct
```

Note: For historical reasons the `direct` deploy interface is sometimes called `agent`. This is because before the Kilo release `ironic-python-agent` used to only support this deploy interface.

Deploy with custom HTTP servers

The direct deploy interface can also be configured to use with custom HTTP servers set up at ironic conductor nodes, images will be cached locally and made accessible by the HTTP server.

To use this deploy interface with a custom HTTP server, set `image_download_source` to `http` in the `[agent]` section.

```
[agent]
...
image_download_source = http
...
```

This configuration affects *glance* and `file://` images. If you want `http(s)://` images to also be cached and served locally, use instead:

```
[agent]
image_download_source = local
```

Note: This option can also be set per node in `driver_info`:

```
baremetal node set <node> --driver-info image_download_source=local
```

or per instance in `instance_info`:

```
baremetal node set <node> --instance-info image_download_source=local
```

You need to set up a workable HTTP server at each conductor node which with `direct` deploy interface enabled, and check `http` related options in the ironic configuration file to match the HTTP server configurations.

```
[deploy]
http_url = http://example.com
http_root = /httpboot
```

Note: See also: *Deploying outside of the provisioning network*.

Each HTTP server should be configured to follow symlinks for images accessible from HTTP service. Please refer to configuration option `FollowSymLinks` if you are using Apache HTTP server, or `disable_symlinks` if Nginx HTTP server is in use.

Ansible deploy

This interface is similar to `direct` in the sense that the image is downloaded by the ramdisk directly from the image store (not from `ironic-conductor` host), but the logic of provisioning the node is held in a set of Ansible playbooks that are applied by the `ironic-conductor` service handling the node. While somewhat more complex to set up, this deploy interface provides greater flexibility in terms of advanced node preparation during provisioning.

This interface is supported by most but not all hardware types declared in `ironic`. However this deploy interface is not enabled by default. To enable it, add `ansible` to the list of enabled deploy interfaces in `enabled_deploy_interfaces` option in the `[DEFAULT]` section of `ironics` configuration file:

```
[DEFAULT]
...
enabled_deploy_interfaces = direct,ansible
...
```

Once enabled, you can specify this deploy interface when creating or updating a node:

```
baremetal node create --driver ipmi --deploy-interface ansible
baremetal node set <NODE> --deploy-interface ansible
```

For more information about this deploy interface, its features and how to use it, see *Ansible deploy interface*.

Ansible deploy interface

`Ansible` is a mature and popular automation tool, written in Python and requiring no agents running on the node being configured. All communications with the node are by default performed over secure SSH transport.

The `ansible` deploy interface uses Ansible playbooks to define the deployment logic. It is not based on `Ironic Python Agent (IPA)` and does not generally need IPA to be running in the deploy ramdisk.

Overview

The main advantage of this deploy interface is extended flexibility in regards to changing and adapting node deployment logic for specific use cases, via Ansible tooling that is already familiar to operators.

It can be used to shorten the usual feature development cycle of

- implementing logic in `ironic`,
- implementing logic in `IPA`,
- rebuilding deploy ramdisk,
- uploading deploy ramdisk to Glance/HTTP storage,
- reassigning deploy ramdisk to nodes,
- restarting `ironic-conductor` service(s) and
- running a test deployment

by using a stable deploy ramdisk and not requiring ironic-conductor restarts (see *Extending playbooks*).

The main disadvantage of this deploy interface is the synchronous manner of performing deployment/cleaning tasks. A separate `ansible-playbook` process is spawned for each node being provisioned or cleaned, which consumes one thread from the thread pool available to the `ironic-conductor` process and blocks this thread until the node provisioning or cleaning step is finished or fails. This has to be taken into account when planning an ironic deployment that enables this deploy interface.

Each action (deploy, clean) is described by a single playbook with roles, which is run whole during deployment, or tag-wise during cleaning. Control of cleaning steps is through tags and auxiliary clean steps file. The playbooks for actions can be set per-node, as can the clean steps file.

Features

Similar to deploy interfaces relying on [Ironic Python Agent \(IPA\)](#), this deploy interface also depends on the deploy ramdisk calling back to ironic APIs `heartbeat` endpoint.

However, the driver is currently synchronous, so only the first heartbeat is processed and is used as a signal to start `ansible-playbook` process.

User images

Supports whole-disk images and partition images:

- compressed images are downloaded to RAM and converted to disk device;
- raw images are streamed to disk directly.

For partition images the driver will create root partition, and, if requested, ephemeral and swap partitions as set in nodes `instance_info` by the Compute service or operator. The create partition table will be of `msdos` type by default, the nodes `disk_label_capability` is honored if set in nodes `instance_info` (see also *Choosing the disk label*).

Configdrive partition

Creating a configdrive partition is supported for both whole disk and partition images, on both `msdos` and GPT labeled disks.

Root device hints

Root device hints are currently supported in their basic form only, with exact matches (see *Specifying the disk for deployment (root device hints)* for more details). If no root device hint is provided for the node, the first device returned as part of `ansible_devices` fact is used as root device to create partitions on or write the whole disk image to.

Node cleaning

Cleaning is supported, both automated and manual. The driver has two default clean steps:

- wiping device metadata
- disk shredding

Their priority can be overridden via `[deploy]\erase_devices_metadata_priority` and `[deploy]\erase_devices_priority` options, respectively, in the ironic configuration file.

As in the case of this driver all cleaning steps are known to the ironic-conductor service, booting the deploy ramdisk is completely skipped when there are no cleaning steps to perform.

Note: Aborting cleaning steps is not supported.

Logging

Logging is implemented as custom Ansible callback module, that makes use of `oslo.log` and `oslo.config` libraries and can re-use logging configuration defined in the main ironic configuration file to set logging for Ansible events, or use a separate file for this purpose.

It works best when `journald` support for logging is enabled.

Requirements

Ansible Tested with, and targets, Ansible 2.5.x

Bootstrap image requirements

- password-less sudo permissions for the user used by Ansible
- python 2.7.x
- openssh-server
- GNU coreutils
- utils-linux
- parted
- gdisk
- qemu-utils
- python-requests (for ironic callback and streaming image download)
- python-netifaces (for ironic callback)

A set of scripts to build a suitable deploy ramdisk based on TinyCore Linux and `tinyipa` ramdisk, and an element for `diskimage-builder` can be found in `ironic-staging-drivers` project but will be eventually migrated to the new `ironic-python-agent-builder` project.

Setting up your environment

1. Install `ironic` (either as part of OpenStack or standalone)
 - If using `ironic` as part of OpenStack, ensure that the Image service is configured to use the Object Storage service as backend, and the Bare Metal service is configured accordingly, see [Configure the Image service for temporary URLs](#).
2. Install Ansible version as specified in `ironic/driver-requirements.txt` file
3. Edit `ironic` configuration file
 - A. Add `ansible` to the list of deploy interfaces defined in `[DEFAULT]\enabled_deploy_interfaces` option.
 - B. Ensure that a hardware type supporting `ansible` deploy interface is enabled in `[DEFAULT]\enabled_hardware_types` option.
 - C. Modify options in the `[ansible]` section of `ironics` configuration file if needed (see [Configuration file](#)).
4. (Re)start `ironic-conductor` service
5. Build suitable deploy kernel and ramdisk images
6. Upload them to Glance or put in your HTTP storage
7. Create new or update existing nodes to use the enabled driver of your choice and populate [Driver properties for the Node](#) when different from defaults.
8. Deploy the node as usual.

Ansible-deploy options

Configuration file

Driver options are configured in `[ansible]` section of `ironic` configuration file, for their descriptions and default values please see [configuration file sample](#).

Driver properties for the Node

Set them per-node via `baremetal node set` command, for example:

```
baremetal node set <node> \
  --deploy-interface ansible \
  --driver-info ansible_username=stack \
  --driver-info ansible_key_file=/etc/ironic/id_rsa
```

ansible_username User name to use for Ansible to access the node. Default is taken from `[ansible]/default_username` option of the `ironic` configuration file (defaults to `ansible`).

ansible_key_file Private SSH key used to access the node. Default is taken from `[ansible]/default_key_file` option of the `ironic` configuration file. If neither is set, the default private SSH keys of the user running the `ironic-conductor` process will be used.

ansible_deploy_playbook Playbook to use when deploying this node. Default is taken from [ansible]/default_deploy_playbook option of the ironic configuration file (defaults to deploy.yaml).

ansible_shutdown_playbook Playbook to use to gracefully shutdown the node in-band. Default is taken from [ansible]/default_shutdown_playbook option of the ironic configuration file (defaults to shutdown.yaml).

ansible_clean_playbook Playbook to use when cleaning the node. Default is taken from [ansible]/default_clean_playbook option of the ironic configuration file (defaults to clean.yaml).

ansible_clean_steps_config Auxiliary YAML file that holds description of cleaning steps used by this node, and defines playbook tags in ansible_clean_playbook file corresponding to each cleaning step. Default is taken from [ansible]/default_clean_steps_config option of the ironic configuration file (defaults to clean_steps.yaml).

ansible_python_interpreter Absolute path to the python interpreter on the managed machine. Default is taken from [ansible]/default_python_interpreter option of the ironic configuration file. Ansible uses /usr/bin/python by default.

Customizing the deployment logic

Expected playbooks directory layout

The [ansible]\playbooks_path option in the ironic configuration file is expected to have a standard layout for an Ansible project with some additions:

```
<playbooks_path>
|
| \_ inventory
| \_ add-ironic-nodes.yaml
| \_ roles
|   \_ role1
|   \_ role2
|   \_ ...
|
| \_ callback_plugins
|   \_ ...
|
| \_ library
|   \_ ...
```

The extra files relied by this driver are:

inventory Ansible inventory file containing a single entry of conductor ansible_connection=local. This basically defines an alias to localhost. Its purpose is to make logging for tasks performed by Ansible locally and referencing the localhost in playbooks more intuitive. This also suppresses warnings produced by Ansible about hosts file being empty.

add-ironic-nodes.yaml This file contains an Ansible play that populates in-memory Ansible inventory with access information received from the ansible-deploy interface, as well as some per-node variables. Include it in all your custom playbooks as the first play.

The default `deploy.yaml` playbook is using several smaller roles that correspond to particular stages of deployment process:

- `discover` - e.g. set root device and image target
- `prepare` - if needed, prepare system, for example create partitions
- `deploy` - download/convert/write user image and configdrive
- `configure` - post-deployment steps, e.g. installing the bootloader

Some more included roles are:

- `shutdown` - used to gracefully power the node off in-band
- `clean` - defines cleaning procedure, with each clean step defined as separate playbook tag.

Extending playbooks

Most probably youd start experimenting like this:

1. Create a copy of `deploy.yaml` playbook *in the same folder*, name it distinctively.
2. Create Ansible roles with your customized logic in `roles` folder.
 - A. In your custom `deploy` playbook, replace the `prepare` role with your own one that defines steps to be run *before* image download/writing. This is a good place to set facts overriding those provided/omitted by the driver, like `ironic_partitions` or `ironic_root_device`, and create custom partitions or (software) RAIDs.
 - B. In your custom `deploy` playbook, replace the `configure` role with your own one that defines steps to be run *after* image is written to disk. This is a good place for example to configure the bootloader and add kernel options to avoid additional reboots.
 - C. Use those new roles in your new playbook.
3. Assign the custom `deploy` playbook youve created to the nodes `driver_info/ansible_deploy_playbook` field.
4. Run deployment.
 - A. No `ironic-conductor` restart is necessary.
 - B. A new `deploy ramdisk` must be built and assigned to nodes only when you want to use a command/script/package not present in the current `deploy ramdisk` and you can not or do not want to install those at runtime.

Variables you have access to

This driver will pass the single JSON-ified extra var argument to Ansible (as in `ansible-playbook -e . .`). Those values are then accessible in your plays as well (some of them are optional and might not be defined):

```
ironic:
  nodes:
  - ip: "<IPADDRESS>"
    name: "<NODE_UUID>"
```

(continues on next page)

(continued from previous page)

```

user: "<USER ANSIBLE WILL USE>"
extra: "<COPY OF NODE's EXTRA FIELD>"
image:
  url: "<URL TO FETCH THE USER IMAGE FROM>"
  disk_format: "<qcow2|raw|...>"
  container_format: "<bare|...>"
  checksum: "<hash-algo:hashstring>"
  mem_req: "<REQUIRED FREE MEMORY TO DOWNLOAD IMAGE TO RAM>"
  tags: "<LIST OF IMAGE TAGS AS DEFINED IN GLANCE>"
  properties: "<DICT OF IMAGE PROPERTIES AS DEFINED IN GLANCE>"
configdrive:
  type: "<url|file>"
  location: "<URL OR PATH ON CONDUCTOR>"
partition_info:
  label: "<msdos|gpt>"
  preserve_ephemeral: "<bool>"
  ephemeral_format: "<FILESYSTEM TO CREATE ON EPHEMERAL PARTITION>"
  partitions: "<LIST OF PARTITIONS IN FORMAT EXPECTED BY PARTED MODULE>"
raid_config: "<COPY OF NODE's TARGET_RAID_CONFIG FIELD>"

```

ironic.nodes List of dictionaries (currently of only one element) that will be used by `add-ironic-nodes.yaml` play to populate in-memory inventory. It also contains a copy of nodes `extra` field so you can access it in the playbooks. The Ansibles host is set to nodes UUID.

ironic.image All fields of nodes `instance_info` that start with `image_` are passed inside this variable. Some extra notes and fields:

- `mem_req` is calculated from image size (if available) and config option `[ansible]extra_memory`.
- if `checksum` is not in the form `<hash-algo>:<hash-sum>`, hashing algorithm is assumed to be md5 (default in Glance).
- `validate_certs` - boolean (yes/no) flag that turns validating image store SSL certificate on or off (default is yes). Governed by `[ansible]image_store_insecure` option in ironic configuration file.
- `cafile` - custom CA bundle to use for validating image store SSL certificate. Takes value of `[ansible]image_store_cafile` if that is defined. Currently is not used by default playbooks, as Ansible has no way to specify the custom CA bundle to use for single HTTPS actions, however you can use this value in your custom playbooks to for example upload and register this CA in the ramdisk at deploy time.
- `client_cert` - cert file for client-side SSL authentication. Takes value of `[ansible]image_store_certfile` option if defined. Currently is not used by default playbooks, however you can use this value in your custom playbooks.
- `client_key` - private key file for client-side SSL authentication. Takes value of `[ansible]image_store_keyfile` option if defined. Currently is not used by default playbooks, however you can use this value in your custom playbooks.

ironic.partition_info.partitions Optional. List of dictionaries defining partitions to create on the node in the form:


```

partitions:
- name: "<NAME OF PARTITION>"
  unit: "<UNITS FOR SIZE>"
  size: "<SIZE OF THE PARTITION>"
  type: "<primary|extended|logical>"
  align: "<ONE OF PARTED-SUPPORTED OPTIONS>"
  format: "<PARTITION TYPE TO SET>"
  flags:
    flag_name: "<bool>"

```

The driver will populate this list from `root_gb`, `swap_mb` and `ephemeral_gb` fields of `instance_info`. The driver will also prepend the `bios_grub`-labeled partition when deploying on GPT-labeled disk, and pre-create a 64 MiB partition for `configdrive` if it is set in `instance_info`.

Please read the documentation included in the `ironic_parted` modules source for more info on the module and its arguments.

ironic.partition_info.ephemeral_format Optional. Taken from `instance_info`, it defines file system to be created on the ephemeral partition. Defaults to the value of `[pxe]\default_ephemeral_format` option in `ironic` configuration file.

ironic.partition_info.preserve_ephemeral Optional. Taken from the `instance_info`, it specifies if the ephemeral partition must be preserved or rebuilt. Defaults to `no`.

ironic.raid_config Taken from the `target_raid_config` if not empty, it specifies the RAID configuration to apply.

As usual for Ansible playbooks, you also have access to standard Ansible facts discovered by `setup` module.

Included custom Ansible modules

The provided `playbooks_path/library` folder includes several custom Ansible modules used by default implementation of `deploy` and `prepare` roles. You can use these modules in your playbooks as well.

stream_url Streaming download from HTTP(S) source to the disk device directly, tries to be compatible with Ansibles `get_url` module in terms of module arguments. Due to the low level of such operation it is not idempotent.

ironic_parted creates partition tables and partitions with `parted` utility. Due to the low level of such operation it is not idempotent. Please read the documentation included in the modules source for more information about this module and its arguments. The name is chosen so that the `parted` module included in Ansible is not shadowed.

Ramdisk deploy

The ramdisk interface is intended to provide a mechanism to deploy an instance where the item to be deployed is in reality a ramdisk. It is documented separately, see *Booting a Ramdisk or an ISO*.

Custom agent deploy

The custom-agent deploy interface is designed for operators who want to completely orchestrate writing the instance image using [in-band deploy steps](#) from a [custom agent image](#). If you use this deploy interface, you are responsible to provide all necessary deploy steps with priorities between 61 and 99 (see *Agent steps* for information on priorities).

4.1.2 Hardware Types

iBMC driver

Overview

The `ibmc` driver is targeted for Huawei V5 series rack server such as 2288H V5, CH121 V5. The `iBMC` hardware type enables the user to take advantage of features of [Huawei iBMC](#) to control Huawei server.

The `ibmc` hardware type supports the following Ironic interfaces:

- Management Interface: Boot device management
- Power Interface: Power management
- *RAID Interface*: RAID controller and disk management
- *Vendor Interface*: `ibmc` passthru interfaces

Prerequisites

The HUAWEI iBMC Client library should be installed on the `ironic conductor` node(s).

For example, it can be installed with `pip`:

```
sudo pip install python-ibmcclient
```

Enabling the iBMC driver

1. Add `ibmc` to the list of `enabled_hardware_types`, `enabled_power_interfaces`, `enabled_vendor_interfaces` and `enabled_management_interfaces` in `/etc/ironic/ironic.conf`. For example:

```
[DEFAULT]
...
enabled_hardware_types = ibmc
enabled_power_interfaces = ibmc
```

(continues on next page)

(continued from previous page)

```
enabled_management_interfaces = ibmc
enabled_raid_interfaces = ibmc
enabled_vendor_interfaces = ibmc
```

2. Restart the ironic conductor service:

```
sudo service ironic-conductor restart

# Or, for RDO:
sudo systemctl restart openstack-ironic-conductor
```

Registering a node with the iBMC driver

Nodes configured to use the driver should have the `driver` property set to `ibmc`.

The following properties are specified in the nodes `driver_info` field:

- `ibmc_address`:

The URL address to the ibmc controller. It must include the authority portion of the URL, and can optionally include the scheme. If the scheme is missing, `https` is assumed. For example: <https://ibmc.example.com>. This is required.

- `ibmc_username`:

User account with admin/server-profile access privilege. This is required.

- `ibmc_password`:

User account password. This is required.

- `ibmc_verify_ca`:

If `ibmc_address` has the **https** scheme, the driver will use a secure (TLS) connection when talking to the ibmc controller. By default (if this is set to `True`), the driver will try to verify the host certificates. This can be set to the path of a certificate file or directory with trusted certificates that the driver will use for verification. To disable verifying TLS, set this to `False`. This is optional.

The `baremetal node create` command can be used to enroll a node with the `ibmc` driver. For example:

```
baremetal node create --driver ibmc
--driver-info ibmc_address=https://example.com \
--driver-info ibmc_username=admin \
--driver-info ibmc_password=password
```

For more information about enrolling nodes see *Enrollment* in the install guide.

RAID Interface

Currently, only RAID controller which supports OOB management can be managed.

See *RAID Configuration* for more information on Ironic RAID support.

The following properties are supported by the iBMC raid interface implementation, `ibmc`:

Mandatory properties

- `size_gb`: Size in gigabytes (integer) for the logical disk. Use `MAX` as `size_gb` if this logical disk is supposed to use the rest of the space available.
- `raid_level`: RAID level for the logical disk. Valid values are JBOD, 0, 1, 5, 6, 1+0, 5+0 and 6+0. And it is possible that some RAID controllers can only support a subset RAID levels.

Note: RAID level 2 is not supported by iBMC driver.

Optional properties

- `is_root_volume`: Optional. Specifies whether this disk is a root volume. By default, this is `False`.
- `volume_name`: Optional. Name of the volume to be created. If this is not specified, it will be `N/A`.

Backing physical disk hints

See *RAID Configuration* for more information on backing disk hints.

These are machine-independent properties. The hints are specified for each logical disk to help Ironic find the desired disks for RAID configuration.

- `share_physical_disks`
- `disk_type`
- `interface_type`
- `number_of_physical_disks`

Backing physical disks

These are HUAWEI RAID controller dependent properties:

- `controller`: Optional. Supported values are: RAID storage id, RAID storage name or RAID controller name. If a bare metal server have more than one controller, this is mandatory. Typical values would look like:
 - RAID Storage Id: `RAIDStorage0`
 - RAID Storage Name: `RAIDStorage0`

- RAID Controller Name: RAID Card1 Controller.
- `physical_disks`: Optional. Supported values are: disk-id, disk-name or disk serial number. Typical values for hdd disk would look like:
 - Disk Id: HDDPlaneDisk0
 - Disk Name: Disk0.
 - Disk SerialNumber: 38DGK77LF77D

Delete RAID configuration

For `delete_configuration` step, `ibmc` will do:

- delete all logical disks
- delete all hot-spare disks

Logical disks creation priority

Logical Disks creation priority based on three properties:

- `share_physical_disks`
- `physical_disks`
- `size_gb`

The logical disks creation priority strictly follow the table below, if multiple logical disks have the same priority, then they will be created with the same order in `logical_disks` array.

Share physical disks	Specified Physical Disks	Size
no	yes	int max
no	no	int
yes	yes	int
yes	yes	max
yes	no	int
yes	no	max
no	no	max

Physical disks choice strategy

Note: `physical-disk-group`: a group of physical disks which have been used by some logical-disks with same RAID level.

- If no `physical_disks` are specified, the waste least strategy will be used to choose the physical disks.
 - waste least disk capacity: when using disks with different capacity, it will cause a waste of disk capacity. This is to avoid with highest priority.

- using least total disk capacity: for example, we can create 400G RAID 5 with both 5 100G-disks and 3 200G-disks. 5 100G disks is a better strategy because it uses a 500G capacity totally. While 3 200G-disks are 600G totally.
- using least disk count: finally, if waste capacity and total disk capacity are both the same (it rarely happens?), we will choose the one with the minimum number of disks.
- when `share_physical_disks` option is present, `ibmc` driver will create logical disk upon existing physical-disk-group list first. Only when no existing physical-disk-group matches, then it chooses unused physical disks with same strategy described above. When multiple exists physical-disk-groups matches, it will use waste least strategy too, the bigger capacity left the better. For example, to create a logical disk shown below on a `ibmc` server which has two RAID5 logical disks already. And the shareable capacity of this two logical-disks are 500G and 300G, then `ibmc` driver will choose the second one.

```
{
  "logical_disks": [
    {
      "controller": "RAID Card1 Controller",
      "raid_level": "5",
      "size_gb": 100,
      "share_physical_disks": true
    }
  ]
}
```

And the `ibmc` server has two RAID5 logical disks already.

- When `size_gb` is set to `MAX`, `ibmc` driver will auto work through all possible cases and choose the best solution which has the biggest capacity and use least capacity. For example: to create a RAID 5+0 logical disk with `MAX` size in a server has 9 200G-disks, it will finally choose 8 disks + span-number 2 but not 9 disks + span-number 3. Although they both have 1200G capacity totally, but the former uses only 8 disks and the latter uses 9 disks. If you want to choose the latter solution, you can specified the disk count to use by adding `number_of_physical_disks` option.

```
{
  "logical_disks": [
    {
      "controller": "RAID Card1 Controller",
      "raid_level": "5+0",
      "size_gb": "MAX"
    }
  ]
}
```

Examples

In a typical scenario we may want to create:

- RAID 5, 500G, root OS volume with 3 disks
- RAID 5, rest available space, data volume with rest disks

```
{
  "logical_disks": [
    {
      "volume_name": "os_volume",
      "controller": "RAID Card1 Controller",
      "is_root_volume": "True",
      "physical_disks": [
        "Disk0",
        "Disk1",
        "Disk2"
      ],
      "raid_level": "5",
      "size_gb": "500"
    },
    {
      "volume_name": "data_volume",
      "controller": "RAID Card1 Controller",
      "raid_level": "5",
      "size_gb": "MAX"
    }
  ]
}
```

Vendor Interface

The `ibmc` hardware type provides vendor passthru interfaces shown below:

Method Name	HTTP Method	Description
<code>boot_up_seq</code>	GET	Query boot up sequence
<code>get_raid_controller_list</code>	GET	Query RAID controller summary info

iDRAC driver

Overview

The integrated Dell Remote Access Controller (**iDRAC**) is an out-of-band management platform on Dell EMC servers, and is supported directly by the `idrac` hardware type. This driver uses the Dell Web Services for Management (WSMAN) protocol and the standard Distributed Management Task Force (DMTF) Redfish protocol to perform all of its functions.

`iDRAC` hardware is also supported by the generic `ipmi` and `redfish` hardware types, though with smaller feature sets.

Key features of the Dell iDRAC driver include:

- Out-of-band node inspection
- Boot device management and firmware management
- Power management
- RAID controller management and RAID volume configuration
- BIOS settings configuration

Ironic Features

The `idrac` hardware type supports the following Ironic interfaces:

- *BIOS Interface*: BIOS management
- *Inspect Interface*: Hardware inspection
- *Management Interface*: Boot device and firmware management
- *Power Interface*: Power management
- *RAID Interface*: RAID controller and disk management
- *Vendor Interface*: BIOS management (WSMAN) and eject virtual media (Redfish)

Prerequisites

The `idrac` hardware type requires the `python-dracclient` library to be installed on the ironic conductor node(s) if an Ironic node is configured to use an `idrac-wsman` interface implementation, for example:

```
sudo pip install 'python-dracclient>=3.1.0'
```

Additionally, the `idrac` hardware type requires the `sushy` library to be installed on the ironic conductor node(s) if an Ironic node is configured to use an `idrac-redfish` interface implementation, for example:

```
sudo pip install 'python-dracclient>=3.1.0' 'sushy>=2.0.0'
```

Enabling

The iDRAC driver supports WSMAN for the bios, inspect, management, power, raid, and vendor interfaces. In addition, it supports Redfish for the bios, inspect, management, power, and raid interfaces. The iDRAC driver allows you to mix and match WSMAN and Redfish interfaces.

The `idrac-wsman` implementation must be enabled to use WSMAN for an interface. The `idrac-redfish` implementation must be enabled to use Redfish for an interface.

To enable the `idrac` hardware type with the minimum interfaces, all using WSMAN, add the following to your `/etc/ironic/ironic.conf`:

[DEFAULT]

```
enabled_hardware_types=idrac
enabled_management_interfaces=idrac-wsman
enabled_power_interfaces=idrac-wsman
```

To enable all optional features (BIOS, inspection, RAID, and vendor passthru) using Redfish where it is supported and WSMAN where not, use the following configuration:

[DEFAULT]

```
enabled_hardware_types=idrac
enabled_bios_interfaces=idrac-redfish
enabled_inspect_interfaces=idrac-redfish
enabled_management_interfaces=idrac-redfish
enabled_power_interfaces=idrac-redfish
enabled_raid_interfaces=idrac-redfish
enabled_vendor_interfaces=idrac-redfish
```

Below is the list of supported interface implementations in priority order:

Interface	Supported Implementations
bios	idrac-wsman, idrac-redfish, no-bios
boot	ipxe, pxe, idrac-redfish-virtual-media
console	no-console
deploy	direct, ansible, ramdisk
inspect	idrac-wsman, idrac, idrac-redfish, inspector, no-inspect
management	idrac-wsman, idrac, idrac-redfish
network	flat, neutron, noop
power	idrac-wsman, idrac, idrac-redfish
raid	idrac-wsman, idrac, idrac-redfish, no-raid
rescue	no-rescue, agent
storage	noop, cinder, external
vendor	idrac-wsman, idrac, idrac-redfish, no-vendor

Note: idrac is the legacy name of the WSMAN interface. It has been deprecated in favor of idrac-wsman and may be removed in a future release.

Protocol-specific Properties

The WSMAN and Redfish protocols require different properties to be specified in the Ironic nodes `driver_info` field to communicate with the bare metal systems iDRAC.

The WSMAN protocol requires the following properties:

- `drac_username`: The WSMAN user name to use when communicating with the iDRAC. Usually `root`.
- `drac_password`: The password for the WSMAN user to use when communicating with the iDRAC.

- `drac_address`: The IP address of the iDRAC.

The Redfish protocol requires the following properties:

- `redfish_username`: The Redfish user name to use when communicating with the iDRAC. Usually `root`.
- `redfish_password`: The password for the Redfish user to use when communicating with the iDRAC.
- `redfish_address`: The URL address of the iDRAC. It must include the authority portion of the URL, and can optionally include the scheme. If the scheme is missing, `https` is assumed.
- `redfish_system_id`: The Redfish ID of the server to be managed. This should always be: `/redfish/v1/Systems/System.Embedded.1`.

For other Redfish protocol parameters see *Redfish driver*.

If using only interfaces which use WSMAN (`idrac-wsman`), then only the WSMAN properties must be supplied. If using only interfaces which use Redfish (`idrac-redfish`), then only the Redfish properties must be supplied. If using a mix of interfaces, where some use WSMAN and others use Redfish, both the WSMAN and Redfish properties must be supplied.

Enrolling

The following command enrolls a bare metal node with the `idrac` hardware type using WSMAN for all interfaces:

```
baremetal node create --driver idrac \  
  --driver-info drac_username=user \  
  --driver-info drac_password=pa$$w0rd \  
  --driver-info drac_address=drac.host
```

The following command enrolls a bare metal node with the `idrac` hardware type using Redfish for all interfaces:

```
baremetal node create --driver idrac \  
  --driver-info redfish_username=user \  
  --driver-info redfish_password=pa$$w0rd \  
  --driver-info redfish_address=drac.host \  
  --driver-info redfish_system_id=/redfish/v1/Systems/System.Embedded.1 \  
  --bios-interface idrac-redfish \  
  --inspect-interface idrac-redfish \  
  --management-interface idrac-redfish \  
  --power-interface idrac-redfish \  
  --raid-interface idrac-redfish \  
  --vendor-interface idrac-redfish
```

The following command enrolls a bare metal node with the `idrac` hardware type assuming a mix of Redfish and WSMAN interfaces are used:

```
baremetal node create --driver idrac \  
  --driver-info drac_username=user \  
  --driver-info drac_password=pa$$w0rd
```

(continues on next page)

(continued from previous page)

```

--driver-info drac_address=drac.host \
--driver-info redfish_username=user \
--driver-info redfish_password=password \
--driver-info redfish_address=drac.host \
--driver-info redfish_system_id=/redfish/v1/Systems/System.Embedded.1 \
--bios-interface idrac-redfish \
--inspect-interface idrac-redfish \
--management-interface idrac-redfish \
--power-interface idrac-redfish

```

Note: If using WSMAN for the management interface, then WSMAN must be used for the power interface. The same applies to Redfish. It is currently not possible to use Redfish for one and WSMAN for the other.

BIOS Interface

The BIOS interface implementations supported by the idrac hardware type allows BIOS to be configured with the standard clean/deploy step approach.

Example

A clean step to enable Virtualization and SRIOV in BIOS of an iDRAC BMC would be as follows:

```

{
  "target": "clean",
  "clean_steps": [
    {
      "interface": "bios",
      "step": "apply_configuration",
      "args": {
        "settings": [
          {
            "name": "ProcVirtualization",
            "value": "Enabled"
          },
          {
            "name": "SriovGlobalEnable",
            "value": "Enabled"
          }
        ]
      }
    }
  ]
}

```

See the *Known Issues* for a known issue with factory_reset clean step. For additional details of BIOS configuration, see *BIOS Configuration*.

Inspect Interface

The Dell iDRAC out-of-band inspection process catalogs all the same attributes of the server as the IPMI driver. Unlike IPMI, it does this without requiring the system to be rebooted, or even to be powered on. Inspection is performed using the Dell WSMAN or Redfish protocol directly without affecting the operation of the system being inspected.

The inspection discovers the following properties:

- `cpu_arch`: cpu architecture
- `cpus`: number of cpus
- `local_gb`: disk size in gigabytes
- `memory_mb`: memory size in megabytes

Extra capabilities:

- `boot_mode`: UEFI or BIOS boot mode.
- `pci_gpu_devices`: number of GPU devices connected to the bare metal.

It also creates baremetal ports for each NIC port detected in the system. The `idrac-wsman inspect` interface discovers which NIC ports are configured to PXE boot and sets `pxe_enabled` to `True` on those ports. The `idrac-redfish inspect` interface does not currently set `pxe_enabled` on the ports. The user should ensure that `pxe_enabled` is set correctly on the ports following inspection with the `idrac-redfish inspect` interface.

Management Interface

The management interface for `idrac-redfish` supports updating firmware on nodes using a manual cleaning step.

See *Redfish driver* for more information on firmware update support.

RAID Interface

See *RAID Configuration* for more information on Ironic RAID support.

The following properties are supported by the iDRAC WSMAN and Redfish RAID interface implementation:

Note: When using `idrac-redfish` for RAID interface iDRAC firmware greater than 4.40.00.00 is required.

Mandatory properties

- `size_gb`: Size in gigabytes (integer) for the logical disk. Use `MAX` as `size_gb` if this logical disk is supposed to use the rest of the space available.
- `raid_level`: RAID level for the logical disk. Valid values are `0`, `1`, `5`, `6`, `1+0`, `5+0` and `6+0`.

Note: JBOD and 2 are not supported, and will fail with reason: Cannot calculate spans for RAID level.

Optional properties

- `is_root_volume`: Optional. Specifies whether this disk is a root volume. By default, this is `False`.
- `volume_name`: Optional. Name of the volume to be created. If this is not specified, it will be auto-generated.

Backing physical disk hints

Note: Backing physical disk hints are not widely tested with `idrac-redfish` yet and they might not work as desired. This will be addressed in future releases.

See *RAID Configuration* for more information on backing disk hints.

These are machine-independent information. The hints are specified for each logical disk to help Ironic find the desired disks for RAID configuration.

- `disk_type`
- `interface_type`
- `share_physical_disks`
- `number_of_physical_disks`

Backing physical disks

These are Dell RAID controller-specific values and must match the names provided by the iDRAC.

- `controller`: Mandatory. The name of the controller to use.
- `physical_disks`: Optional. The names of the physical disks to use.

Note: `physical_disks` is a mandatory parameter if the property `size_gb` is set to `MAX`.

Examples

Creation of RAID 1+0 logical disk with six disks on one controller:

```
{ "logical_disks":
  [ { "controller": "RAID.Integrated.1-1",
      "is_root_volume": "True",
      "physical_disks": [
        "Disk.Bay.0:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.1:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.2:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.3:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.4:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.5:Enclosure.Internal.0-1:RAID.Integrated.1-1"],
      "raid_level": "1+0",
      "size_gb": "MAX"}]}
```

Manual RAID Invocation

The following command can be used to delete any existing RAID configuration. It deletes all virtual disks/RAID volumes, unassigns all global and dedicated hot spare physical disks, and clears foreign configuration:

```
baremetal node clean --clean-steps \
  '[{"interface": "raid", "step": "delete_configuration"}]' ${node_uuid}
```

The following command shows an example of how to set the target RAID configuration:

```
baremetal node set --target-raid-config '{ "logical_disks":
  [ { "controller": "RAID.Integrated.1-1",
      "is_root_volume": true,
      "physical_disks": [
        "Disk.Bay.0:Enclosure.Internal.0-1:RAID.Integrated.1-1",
        "Disk.Bay.1:Enclosure.Internal.0-1:RAID.Integrated.1-1"],
      "raid_level": "0",
      "size_gb": "MAX"}]}' ${node_uuid}
```

The following command can be used to create a RAID configuration:

```
baremetal node clean --clean-steps \
  '[{"interface": "raid", "step": "create_configuration"}]' <node>
```

When the physical disk names or controller names are not known, the following Python code example shows how the `python-dracclient` can be used to fetch the information directly from the Dell bare metal:

```
import dracclient.client

client = dracclient.client.DRACClient(
```

(continues on next page)

(continued from previous page)

```

host="192.168.1.1",
username="root",
password="calvin")
controllers = client.list_raid_controllers()
print(controllers)

physical_disks = client.list_physical_disks()
print(physical_disks)

```

Or using sushy with Redfish:

```

import sushy

client = sushy.Sushy('https://192.168.1.1', username='root', password='calvin
→', verify=False)
for s in client.get_system_collection().get_members():
    print("System: %(id)s" % {'id': s.identity})
    for c in system1.storage.get_members():
        print("\tController: %(id)s" % {'id': c.identity})
        for d in c.drives:
            print("\t\tDrive: %(id)s" % {'id': d.identity})

```

Vendor Interface

idrac-wsman

Dell iDRAC BIOS management is available through the Ironic WSMAN vendor passthru interface.

Method Name	HTTP Method	Description
abandon_bios_config	DELETE	Abandon a BIOS configuration job.
commit_bios_config	POST	Commit a BIOS configuration job submitted through <code>set_bios_config</code> . Required argument: <code>reboot</code> - indicates whether a reboot job should be automatically created with the config job. Returns a dictionary containing the <code>job_id</code> key with the ID of the newly created config job, and the <code>reboot_required</code> key indicating whether the node needs to be rebooted to execute the config job.
get_bios_config	GET	Returns a dictionary containing the nodes BIOS settings.
list_unfinished_bios_config_jobs	GET	Returns a dictionary containing the key <code>unfinished_jobs</code> ; its value is a list of dictionaries. Each dictionary represents an unfinished config job object.
set_bios_config	POST	Change the BIOS configuration on a node. Required argument: a dictionary of <code>{AttributeName: NewValue}</code> . Returns a dictionary containing the <code>is_commit_required</code> key indicating whether <code>commit_bios_config</code> needs to be called to apply the changes and the <code>is_reboot_required</code> value indicating whether the server must also be rebooted. Possible values are <code>true</code> and <code>false</code> .

Examples

Get BIOS Config

```
baremetal node passthru call --http-method GET <node> get_bios_config
```

Snippet of output showing virtualization enabled:

```
{"ProcVirtualization": {  
  "current_value": "Enabled",  
  "instance_id": "BIOS.Setup.1-1:ProcVirtualization",  
  "name": "ProcVirtualization",  
  "pending_value": null,  
  "possible_values": [  
    "Enabled",  
    "Disabled"],  
  "read_only": false }}
```

There are a number of items to note from the above snippet:

- name: this is the name to use in a call to `set_bios_config`.
- current_value: the current state of the setting.
- pending_value: if the value has been set, but not yet committed, the new value is shown here. The change can either be committed or abandoned.
- possible_values: shows a list of valid values which can be used in a call to `set_bios_config`.
- read_only: indicates if the value is capable of being changed.

Set BIOS Config

```
baremetal node passthru call <node> set_bios_config --arg "name=value"
```

Walkthrough of performing a BIOS configuration change:

The following section demonstrates how to change BIOS configuration settings, detect that a commit and reboot are required, and act on them accordingly. The two properties that are being changed are:

- Enable virtualization technology of the processor
- Globally enable SR-IOV

```
baremetal node passthru call <node> set_bios_config \  
--arg "ProcVirtualization=Enabled" \  
--arg "SriovGlobalEnable=Enabled"
```

This returns a dictionary indicating what actions are required next:

```
{  
  "is_reboot_required": true,  
  "is_commit_required": true  
}
```


Commit BIOS Changes

The next step is to commit the pending change to the BIOS. Note that in this example, the `reboot` argument is set to `true`. The response indicates that a reboot is no longer required as it has been scheduled automatically by the `commit_bios_config` call. If the `reboot` argument is not supplied, the job is still created, however it remains in the `scheduled` state until a reboot is performed. The reboot can be initiated through the Ironic power API.

```
baremetal node passthru call <node> commit_bios_config \
  --arg "reboot=true"
```

```
{
  "job_id": "JID_499377293428",
  "reboot_required": false
}
```

The state of any executing job can be queried:

```
baremetal node passthru call --http-method GET <node> list_unfinished_jobs
```

```
{"unfinished_jobs":
  [{"status": "Scheduled",
    "name": "ConfigBIOS:BIOS.Setup.1-1",
    "until_time": "TIME_NA",
    "start_time": "TIME_NOW",
    "message": "Task successfully scheduled.",
    "percent_complete": "0",
    "id": "JID_499377293428"}]}}
```

Abandon BIOS Changes

Instead of committing, a pending change can be abandoned:

```
baremetal node passthru call --http-method DELETE <node> abandon_bios_config
```

The `abandon` command does not provide a response body.

Change Boot Mode

The boot mode of the iDRAC can be changed to:

- BIOS - Also called legacy or traditional boot mode. The BIOS initializes the systems processors, memory, bus controllers, and I/O devices. After initialization is complete, the BIOS passes control to operating system (OS) software. The OS loader uses basic services provided by the system BIOS to locate and load OS modules into system memory. After booting the system, the BIOS and embedded management controllers execute system management algorithms, which monitor and optimize the condition of the underlying hardware. BIOS configuration settings enable fine-tuning of the performance, power management, and reliability features of the system.

- UEFI - The Unified Extensible Firmware Interface does not change the traditional purposes of the system BIOS. To a large extent, a UEFI-compliant BIOS performs the same initialization, boot, configuration, and management tasks as a traditional BIOS. However, UEFI does change the interfaces and data structures the BIOS uses to interact with I/O device firmware and operating system software. The primary intent of UEFI is to eliminate shortcomings in the traditional BIOS environment, enabling system firmware to continue scaling with industry trends.

The UEFI boot mode offers:

- Improved partitioning scheme for boot media
- Support for media larger than 2 TB
- Redundant partition tables
- Flexible handoff from BIOS to OS
- Consolidated firmware user interface
- Enhanced resource allocation for boot device firmware

The boot mode can be changed via the WSMAN vendor passthru interface as follows:

```
baremetal node passthru call <node> set_bios_config \  
  --arg "BootMode=Uefi"  
  
baremetal node passthru call <node> commit_bios_config \  
  --arg "reboot=true"
```

```
baremetal node passthru call <node> set_bios_config \  
  --arg "BootMode=Bios"  
  
baremetal node passthru call <node> commit_bios_config \  
  --arg "reboot=true"
```

idrac-redfish

Through the `idrac-redfish` vendor passthru interface these methods are available:

Method Name	HTTP Method	Description
<code>eject_media</code>	<code>POST</code>	Eject a virtual media device. If no device is provided then all attached devices will be ejected. Optional argument: <code>boot_device</code> - the boot device to eject, either, <code>cd</code> , <code>dvd</code> , <code>usb</code> or <code>floppy</code> .

Known Issues

Nodes go into maintenance mode

After some period of time, nodes managed by the idrac hardware type may go into maintenance mode in Ironic. This issue can be worked around by changing the Ironic power state poll interval to 70 seconds. See `[conductor]sync_power_state_interval` in `/etc/ironic/ironic.conf`.

PXE reset with factory_reset BIOS clean step

When using the `UEFI boot mode`` with non-default PXE interface, the factory reset can cause the PXE interface to be reset to default, which doesn't allow the server to PXE boot for any further operations. This can cause a `clean_failed` state on the node or `deploy_failed` if you attempt to deploy a node after this step. For now, the only solution is for the operator to manually restore the PXE settings of the server for it to PXE boot again, properly. The problem is caused by the fact that with the `UEFI boot mode`, the idrac uses BIOS settings to manage PXE configuration. This is not the case with the `BIOS boot mode` where the PXE configuration is handled as a configuration job on the integrated NIC itself, independently of the BIOS settings.

WSMAN vendor passthru timeout

When iDRAC is not ready and executing WSMAN vendor passthru commands, they take more time as waiting for iDRAC to become ready again and then time out, for example:

```
baremetal node passthru call --http-method GET \  
aed58dca-1b25-409a-a32f-3a817d59e1e0 list_unfinished_jobs  
Timed out waiting for a reply to message ID 547ce7995342418c99ef1ea4a0054572.  
↔ (HTTP 500)
```

To avoid this need to increase timeout for messaging in `/etc/ironic/ironic.conf` and restart Ironic API service.

```
[DEFAULT]  
rpc_response_timeout = 600
```

Timeout when powering off

Some servers might be slow when soft powering off and time out. The default retry count is 6, resulting in 30 seconds timeout (the default retry interval set by `post_deploy_get_power_state_retry_interval` is 5 seconds). To resolve this issue, increase the timeout to 90 seconds by setting the retry count to 18 as follows:

```
[agent]  
post_deploy_get_power_state_retries = 18
```

Unable to mount remote share with iDRAC firmware 4.40.00.00 or newer

When using iDRAC firmware 4.40.00.00 or newer with virtual media boot and new Virtual Console plug-in type eHTML5, there is an error: Unable to mount remote share. This is a known issue that will be fixed in future iDRAC firmware releases. Until then can adjust settings in iDRAC to use plug-in type HTML5. In iDRAC web UI go to Configuration -> Virtual Console and select Plug-in Type to HTML5.

During upgrade to 4.40.00.00 or newer iDRAC firmware eHTML5 is automatically selected if default plug-in type has been used and never changed. Systems that have plug-in type changed will keep selected plug-in type after iDRAC firmware upgrade.

iLO driver

Overview

iLO driver enables to take advantage of features of iLO management engine in HPE ProLiant servers. The `ilo` hardware type is targeted for HPE ProLiant Gen8 and Gen9 systems which have [iLO 4 management engine](#). From **Pike** release `ilo` hardware type supports ProLiant Gen10 systems which have [iLO 5 management engine](#). `ilo5` conforms to [Redfish API](#) and hence hardware type `redfish` (see [Redfish driver](#)) is also an option for this kind of hardware but it lacks the iLO specific features.

For more details and for up-to-date information (like tested platforms, known issues, etc), please check the [iLO driver wiki page](#).

For enabling Gen10 systems and getting detailed information on Gen10 feature support in Ironic please check this [Gen10 wiki section](#).

Hardware type

ProLiant hardware is primarily supported by the `ilo` hardware type. `ilo5` hardware type is only supported on ProLiant Gen10 and later systems. Both hardware can be used with reference hardware type `ipmi` (see [IPMI driver](#)) and `redfish` (see [Redfish driver](#)). For information on how to enable the `ilo` and `ilo5` hardware type, see [Enabling hardware types](#).

Note: Only HPE ProLiant Gen10 servers supports hardware type `redfish`.

Warning: It is important to note that while the HPE Edgeline series of servers may contain iLO adapters, they are known to not be compatible with the `ilo` hardware type. The `redfish` hardware type should be used instead.

The hardware type `ilo` supports following HPE server features:

- *Boot mode support*
- *UEFI Secure Boot Support*
- *Node Cleaning Support*
- *Node Deployment Customization*

- *Hardware Inspection Support*
- *Swiftless deploy for intermediate images*
- *HTTP(S) Based Deploy Support*
- *Support for iLO driver with Standalone Ironic*
- *RAID Support*
- *Disk Erase Support*
- *Initiating firmware update as manual clean step*
- *Smart Update Manager (SUM) based firmware update*
- *Updating security parameters as manual clean step*
- *Update Minimum Password Length security parameter as manual clean step*
- *Update Authentication Failure Logging security parameter as manual clean step*
- *Activating iLO Advanced license as manual clean step*
- *Firmware based UEFI iSCSI boot from volume support*
- *Certificate based validation in iLO*
- *Rescue mode support*
- *Inject NMI support*
- *Soft power operation support*
- *BIOS configuration support*
- *IPv6 support*
- *Layer 3 or DHCP-less ramdisk booting*

Apart from above features hardware type ilo5 also supports following features:

- *Out of Band RAID Support*
- *Out of Band Sanitize Disk Erase Support*
- *Out of Band One Button Secure Erase Support*
- *UEFI-HTTPS Boot support*

Hardware interfaces

The ilo hardware type supports following hardware interfaces:

- **bios** Supports ilo and no-bios. The default is ilo. They can be enabled by using the [DEFAULT]enabled_bios_interfaces option in ironic.conf as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_bios_interfaces = ilo,no-bios
```

- **boot** Supports `ilo-virtual-media`, `ilo-pxe` and `ilo-ipxe`. The default is `ilo-virtual-media`. The `ilo-virtual-media` interface provides security enhanced PXE-less deployment by using iLO virtual media to boot up the bare metal node. The `ilo-pxe` and `ilo-ipxe` interfaces use PXE and iPXE respectively for deployment (just like *PXE boot*). These interfaces do not require iLO Advanced license. They can be enabled by using the `[DEFAULT]enabled_boot_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_boot_interfaces = ilo-virtual-media,ilo-pxe,ilo-ipxe
```

- **console** Supports `ilo` and `no-console`. The default is `ilo`. They can be enabled by using the `[DEFAULT]enabled_console_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_console_interfaces = ilo,no-console
```

Note: To use `ilo` console interface you need to enable iLO feature IPMI/DCMI over LAN Access on `iLO4` and `iLO5` management engine.

- **inspect** Supports `ilo` and `inspector`. The default is `ilo`. They can be enabled by using the `[DEFAULT]enabled_inspect_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_inspect_interfaces = ilo,inspector
```

Note: `Ironic Inspector` needs to be configured to use `inspector` as the inspect interface.

- **management** Supports only `ilo`. It can be enabled by using the `[DEFAULT]enabled_management_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_management_interfaces = ilo
```

- **power** Supports only `ilo`. It can be enabled by using the `[DEFAULT]enabled_power_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_power_interfaces = ilo
```

- **raid** Supports `agent` and `no-raid`. The default is `no-raid`. They can be enabled by using the `[DEFAULT]enabled_raid_interfaces` option in `ironic.conf` as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_raid_interfaces = agent,no-raid
```

- **storage** Supports cinder and noop. The default is noop. They can be enabled by using the [DEFAULT]enabled_storage_interfaces option in ironic.conf as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_storage_interfaces = cinder,noop
```

Note: The storage interface cinder is supported only when corresponding boot interface of the ilo hardware type based node is ilo-pxe or ilo-ipxe. Please refer to *Boot From Volume* for configuring cinder as a storage interface.

- **rescue** Supports agent and no-rescue. The default is no-rescue. They can be enabled by using the [DEFAULT]enabled_rescue_interfaces option in ironic.conf as given below:

```
[DEFAULT]
enabled_hardware_types = ilo
enabled_rescue_interfaces = agent,no-rescue
```

The ilo5 hardware type supports all the ilo interfaces described above, except for boot and raid interfaces. The details of boot and raid interfaces is as under:

- **raid** Supports ilo5 and no-raid. The default is ilo5. They can be enabled by using the [DEFAULT]enabled_raid_interfaces option in ironic.conf as given below:

```
[DEFAULT]
enabled_hardware_types = ilo5
enabled_raid_interfaces = ilo5,no-raid
```

- **boot** Supports ilo-uefi-https apart from the other boot interfaces supported by ilo hardware type. This can be enabled by using the [DEFAULT]enabled_boot_interfaces option in ironic.conf as given below:

```
[DEFAULT]
enabled_hardware_types = ilo5
enabled_boot_interfaces = ilo-uefi-https,ilo-virtual-media
```

The ilo and ilo5 hardware type support all standard deploy and network interface implementations, see *Enabling hardware interfaces* for details.

The following command can be used to enroll a ProLiant node with ilo hardware type:

```
baremetal node create \
  --driver ilo \
  --deploy-interface direct \
  --raid-interface agent \
  --rescue-interface agent \
  --driver-info ilo_address=<ilo-ip-address> \
```

(continues on next page)

(continued from previous page)

```
--driver-info ilo_username=<ilo-username> \  
--driver-info ilo_password=<ilo-password> \  
--driver-info ilo_deploy_iso=<glance-uuid-of-deploy-iso> \  
--driver-info ilo_rescue_iso=<glance-uuid-of-rescue-iso>
```

The following command can be used to enroll a ProLiant node with ilo5 hardware type:

```
baremetal node create \  
  --driver ilo5 \  
  --deploy-interface direct \  
  --raid-interface ilo5 \  
  --rescue-interface agent \  
  --driver-info ilo_address=<ilo-ip-address> \  
  --driver-info ilo_username=<ilo-username> \  
  --driver-info ilo_password=<ilo-password> \  
  --driver-info ilo_deploy_iso=<glance-uuid-of-deploy-iso> \  
  --driver-info ilo_rescue_iso=<glance-uuid-of-rescue-iso>
```

Please refer to *Enabling drivers and hardware types* for detailed explanation of hardware type.

Node configuration

- Each node is configured for ilo and ilo5 hardware type by setting the following ironic node objects properties in `driver_info`:
 - `ilo_address`: IP address or hostname of the iLO.
 - `ilo_username`: Username for the iLO with administrator privileges.
 - `ilo_password`: Password for the above iLO user.
 - `client_port`: (optional) Port to be used for iLO operations if you are using a custom port on the iLO. Default port used is 443.
 - `client_timeout`: (optional) Timeout for iLO operations. Default timeout is 60 seconds.
 - `ca_file`: (optional) CA certificate file to validate iLO.
 - `console_port`: (optional) Nodes UDP port for console access. Any unused port on the ironic conductor node may be used. This is required only when `ilo-console` interface is used.
- The following properties are also required in node objects `driver_info` if `ilo-virtual-media` boot interface is used:
 - `ilo_deploy_iso`: The glance UUID of the deploy ramdisk ISO image.
 - `instance info/ilo_boot_iso` property to be either boot iso Glance UUID or a HTTP(S) URL. This is optional property and is used when `boot_option` is set to `netboot` or `ramdisk`.

Note: When `boot_option` is set to `ramdisk`, the ironic node must be configured to use `ramdisk` deploy interface. See *Ramdisk deploy* for details.

- `ilo_rescue_iso`: The glance UUID of the rescue ISO image. This is optional property and is used when `rescue` interface is set to `agent`.
- The following properties are also required in node objects `driver_info` if `ilo-pxe` or `ilo-ipxe` boot interface is used:
 - `deploy_kernel`: The glance UUID or a HTTP(S) URL of the deployment kernel.
 - `deploy_ramdisk`: The glance UUID or a HTTP(S) URL of the deployment ramdisk.
 - `rescue_kernel`: The glance UUID or a HTTP(S) URL of the rescue kernel. This is optional property and is used when `rescue` interface is set to `agent`.
 - `rescue_ramdisk`: The glance UUID or a HTTP(S) URL of the rescue ramdisk. This is optional property and is used when `rescue` interface is set to `agent`.
- The following properties are also required in node objects `driver_info` if `ilo-uefi-https` boot interface is used for `ilo5` hardware type:
 - `ilo_deploy_kernel`: The glance UUID or a HTTPS URL of the deployment kernel.
 - `ilo_deploy_ramdisk`: The glance UUID or a HTTPS URL of the deployment ramdisk.
 - `ilo_bootloader`: The glance UUID or a HTTPS URL of the bootloader.
 - `ilo_rescue_kernel`: The glance UUID or a HTTPS URL of the rescue kernel. This is optional property and is used when `rescue` interface is set to `agent`.
 - `ilo_rescue_ramdisk`: The glance UUID or a HTTP(S) URL of the rescue ramdisk. This is optional property and is used when `rescue` interface is set to `agent`.

Note: `ilo-uefi-https` boot interface is supported by only `ilo5` hardware type. If the images are not hosted in glance, the references must be HTTPS URLs hosted by secure web-server. This boot interface can be used only when the current boot mode is UEFI.

- The following parameters are mandatory in `driver_info` if `ilo-inspect` inspect interface is used and SNMPv3 inspection (*SNMPv3 Authentication* in [HPE iLO4 User Guide](#)) is desired:
 - `snmp_auth_user` : The SNMPv3 user.
 - `snmp_auth_prot_password` : The auth protocol pass phrase.
 - `snmp_auth_priv_password` : The privacy protocol pass phrase.

The following parameters are optional for SNMPv3 inspection:

- `snmp_auth_protocol` : The Auth Protocol. The valid values are MD5 and SHA. The iLO default value is MD5.
- `snmp_auth_priv_protocol` : The Privacy protocol. The valid values are AES and DES. The iLO default value is DES.

Note: If configuration values for `ca_file`, `client_port` and `client_timeout` are not provided in the `driver_info` of the node, the corresponding config variables defined under `[ilo]` section in `ironic.conf` will be used.

Prerequisites

- `proliantutils` is a python package which contains a set of modules for managing HPE ProLiant hardware.

Install `proliantutils` module on the ironic conductor node. Minimum version required is 2.8.0:

```
$ pip install "proliantutils>=2.8.0"
```

- `ipmitool` command must be present on the service node(s) where `ironic-conductor` is running. On most distros, this is provided as part of the `ipmitool` package. Please refer to [Hardware Inspection Support](#) for more information on recommended version.

Different configuration for ilo hardware type

Glance Configuration

1. Configure Glance image service with its storage backend as Swift.
2. Set a `temp-url` key for Glance user in Swift. For example, if you have configured Glance with user `glance-swift` and tenant as `service`, then run the below command:

```
swift --os-username=service:glance-swift post -m temp-url-  
↪key:mysecretkeyforglance
```

3. Fill the required parameters in the `[glance]` section in `/etc/ironic/ironic.conf`. Normally you would be required to fill in the following details:

```
[glance]  
swift_temp_url_key=mysecretkeyforglance  
swift_endpoint_url=https://10.10.1.10:8080  
swift_api_version=v1  
swift_account=AUTH_51ea2fb400c34c9eb005ca945c0dc9e1  
swift_container=glance
```

The details can be retrieved by running the below command:

```
$ swift --os-username=service:glance-swift stat -v | grep -i url  
  
StorageURL:      http://10.10.1.10:8080/v1/AUTH_  
↪51ea2fb400c34c9eb005ca945c0dc9e1  
Meta Temp-Url-Key: mysecretkeyforglance
```

4. Swift must be accessible with the same admin credentials configured in Ironic. For example, if Ironic is configured with the below credentials in `/etc/ironic/ironic.conf`:

```
[keystone_authtoken]  
admin_password = password  
admin_user = ironic  
admin_tenant_name = service
```

Ensure `auth_version` in `keystone_authtoken` to 2.

Then, the below command should work.:

```
$ swift --os-username ironic --os-password password --os-tenant-name_
↪service --auth-version 2 stat

Account: AUTH_22af34365a104e4689c46400297f00cb
Containers: 2
Objects: 18
Bytes: 1728346241
Objects in policy "policy-0": 18
Bytes in policy "policy-0": 1728346241
Meta Temp-Url-Key: mysecretkeyforglance
X-Timestamp: 1409763763.84427
X-Trans-Id: tx51de96a28f27401eb2833-005433924b
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

5. Restart the Ironic conductor service:

```
$ service ironic-conductor restart
```

Web server configuration on conductor

- The HTTP(S) web server can be configured in many ways. For apache web server on Ubuntu, refer [here](#)
- Following config variables need to be set in `/etc/ironic/ironic.conf`:
 - `use_web_server_for_images` in `[ilo]` section:

```
[ilo]
use_web_server_for_images = True
```

- `http_url` and `http_root` in `[deploy]` section:

```
[deploy]
# Ironic compute node's http root path. (string value)
http_root=/httpboot

# Ironic compute node's HTTP server URL. Example:
# http://192.1.2.3:8080 (string value)
http_url=http://192.168.0.2:8080
```

`use_web_server_for_images`: If the variable is set to false, the `ilo-virtual-media` boot interface uses swift containers to host the intermediate floppy image and the boot ISO. If the variable is set to true, it uses the local web server for hosting the intermediate files. The default value for `use_web_server_for_images` is False.

`http_url`: The value for this variable is prefixed with the generated intermediate files to generate a URL which is attached in the virtual media.

`http_root`: It is the directory location to which ironic conductor copies the intermediate floppy image and the boot ISO.

Note: HTTPS is strongly recommended over HTTP web server configuration for security enhancement. The `ilo-virtual-media` boot interface will send the instances configdrive over an encrypted channel if web server is HTTPS enabled. However for `ilo-uefi-https` boot interface HTTPS webserver is mandatory as this interface only supports HTTPS URLs.

Enable driver

1. Build a deploy ISO (and kernel and ramdisk) image, see *Building or downloading a deploy ramdisk image*
2. See *Glance Configuration* for configuring glance image service with its storage backend as `swift`.
3. Upload this image to Glance:

```
glance image-create --name deploy-ramdisk.iso --disk-format iso --  
↔container-format bare < deploy-ramdisk.iso
```

4. Enable hardware type and hardware interfaces in `/etc/ironic/ironic.conf`:

```
[DEFAULT]  
enabled_hardware_types = ilo  
enabled_bios_interfaces = ilo  
enabled_boot_interfaces = ilo-virtual-media,ilo-pxe,ilo-ipxe  
enabled_power_interfaces = ilo  
enabled_console_interfaces = ilo  
enabled_raid_interfaces = agent  
enabled_management_interfaces = ilo  
enabled_inspect_interfaces = ilo  
enabled_rescue_interfaces = agent
```

5. Restart the ironic conductor service:

```
$ service ironic-conductor restart
```

Optional functionalities for the ilo hardware type

Boot mode support

The hardware type `ilo` supports automatic detection and setting of boot mode (Legacy BIOS or UEFI).

- When boot mode capability is not configured:
 - If config variable `default_boot_mode` in `[ilo]` section of ironic configuration file is set to either `bios` or `uefi`, then iLO driver uses that boot mode for provisioning the baremetal ProLiant servers.
 - If the pending boot mode is set on the node then iLO driver uses that boot mode for provisioning the baremetal ProLiant servers.
 - If the pending boot mode is not set on the node then iLO driver uses `uefi` boot mode for UEFI capable servers and `bios` when UEFI is not supported.

- When boot mode capability is configured, the driver sets the pending boot mode to the configured value.
- Only one boot mode (either uefi or bios) can be configured for the node.
- If the operator wants a node to boot always in uefi mode or bios mode, then they may use capabilities parameter within properties field of an ironic node.

To configure a node in uefi mode, then set capabilities as below:

```
baremetal node set <node> --property capabilities='boot_mode:uefi'
```

Nodes having boot_mode set to uefi may be requested by adding an extra_spec to the nova flavor:

```
nova flavor-key ironic-test-3 set capabilities:boot_mode="uefi"
nova boot --flavor ironic-test-3 --image test-image instance-1
```

If capabilities is used in extra_spec as above, nova scheduler (ComputeCapabilitiesFilter) will match only ironic nodes which have the boot_mode set appropriately in properties/capabilities. It will filter out rest of the nodes.

The above facility for matching in nova can be used in heterogeneous environments where there is a mix of uefi and bios machines, and operator wants to provide a choice to the user regarding boot modes. If the flavor doesnt contain boot_mode then nova scheduler will not consider boot mode as a placement criteria, hence user may get either a BIOS or UEFI machine that matches with user specified flavors.

The automatic boot ISO creation for UEFI boot mode has been enabled in Kilo. The manual creation of boot ISO for UEFI boot mode is also supported. For the latter, the boot ISO for the deploy image needs to be built separately and the deploy images boot_iso property in glance should contain the glance UUID of the boot ISO. For building boot ISO, add iso element to the diskimage-builder command to build the image. For example:

```
disk-image-create ubuntu baremetal iso
```

UEFI Secure Boot Support

The hardware type ilo supports secure boot deploy, see *UEFI secure boot mode* for details.

iLO specific notes:

In UEFI secure boot, digitally signed bootloader should be able to validate digital signatures of kernel during boot process. This requires that the bootloader contains the digital signatures of the kernel. For the ilo-virtual-media boot interface, it is recommended that boot_iso property for user image contains the glance UUID of the boot ISO. If boot_iso property is not updated in glance for the user image, it would create the boot_iso using bootloader from the deploy iso. This boot_iso will be able to boot the user image in UEFI secure boot environment only if the bootloader is signed and can validate digital signatures of user image kernel.

For HPE ProLiant Gen9 servers, one can enroll public key using iLO System Utilities UI. Please refer to section *Accessing Secure Boot options* in *HP UEFI System Utilities User Guide*. One can also refer to white paper on *Secure Boot for Linux on HP ProLiant servers* for additional details.

For more up-to-date information, refer [iLO driver wiki page](#)

Node Cleaning Support

The hardware type `ilo` and `ilo5` supports node cleaning.

For more information on node cleaning, see [Node cleaning](#)

Supported Automated Cleaning Operations

- The automated cleaning operations supported are:
 - `reset_bios_to_default`: Resets system ROM settings to default. By default, enabled with priority 10. This clean step is supported only on Gen9 and above servers.
 - `reset_secure_boot_keys_to_default`: Resets secure boot keys to manufacturers defaults. This step is supported only on Gen9 and above servers. By default, enabled with priority 20 .
 - `reset_ilo_credential`: Resets the iLO password, if `ilo_change_password` is specified as part of nodes `driver_info`. By default, enabled with priority 30.
 - `clear_secure_boot_keys`: Clears all secure boot keys. This step is supported only on Gen9 and above servers. By default, this step is disabled.
 - `reset_ilo`: Resets the iLO. By default, this step is disabled.
 - `erase_devices`: An inband clean step that performs disk erase on all the disks including the disks visible to OS as well as the raw disks visible to Smart Storage Administrator (SSA). This step supports erasing of the raw disks visible to SSA in Proliant servers only with the ramdisk created using `diskimage-builder` from Ocata release. By default, this step is disabled. See [Disk Erase Support](#) for more details.
- For supported in-band cleaning operations, see [In-band vs out-of-band](#).
- All the automated cleaning steps have an explicit configuration option for priority. In order to disable or change the priority of the automated clean steps, respective configuration option for priority should be updated in `ironic.conf`.
- Updating clean step priority to 0, will disable that particular clean step and will not run during automated cleaning.
- Configuration Options for the automated clean steps are listed under `[ilo]` and `[deploy]` section in `ironic.conf`

```
[ilo]
clean_priority_reset_ilo=0
clean_priority_reset_bios_to_default=10
clean_priority_reset_secure_boot_keys_to_default=20
clean_priority_clear_secure_boot_keys=0
clean_priority_reset_ilo_credential=30

[deploy]
erase_devices_priority=0
```

For more information on node automated cleaning, see [Automated cleaning](#)

Supported Manual Cleaning Operations

- The manual cleaning operations supported are:

activate_license: Activates the iLO Advanced license. This is an out-of-band manual cleaning step associated with the management interface. See *Activating iLO Advanced license as manual clean step* for user guidance on usage. Please note that this operation cannot be performed using the `ilo-virtual-media` boot interface as it needs this type of advanced license already active to use virtual media to boot into to start cleaning operation. Virtual media is an advanced feature. If an advanced license is already active and the user wants to overwrite the current license key, for example in case of a multi-server activation key delivered with a flexible-quantity kit or after completing an Activation Key Agreement (AKA), then the driver can still be used for executing this cleaning step.

apply_configuration: Applies given BIOS settings on the node. See *BIOS configuration support*. This step is part of the `bios` interface.

factory_reset: Resets the BIOS settings on the node to factory defaults. See *BIOS configuration support*. This step is part of the `bios` interface.

create_configuration: Applies RAID configuration on the node. See *RAID Configuration* for more information. This step is part of the `raid` interface.

delete_configuration: Deletes RAID configuration on the node. See *RAID Configuration* for more information. This step is part of the `raid` interface.

update_firmware: Updates the firmware of the devices. Also an out-of-band step associated with the management interface. See *Initiating firmware update as manual clean step* for user guidance on usage. The supported devices for firmware update are: `ilo`, `cpld`, `power_pic`, `bios` and `chassis`. Please refer to below table for their commonly used descriptions.

Device	Description
<code>ilo</code>	BMC for HPE ProLiant servers
<code>cpld</code>	System programmable logic device
<code>power_pic</code>	Power management controller
<code>bios</code>	HPE ProLiant System ROM
<code>chassis</code>	System chassis device

Some devices firmware cannot be updated via this method, such as: storage controllers, host bus adapters, disk drive firmware, network interfaces and Onboard Administrator (OA).

update_firmware_sum: Updates all or list of user specified firmware components on the node using Smart Update Manager (SUM). It is an inband step associated with the management interface. See *Smart Update Manager (SUM) based firmware update* for more information on usage.

security_parameters_update: Updates the Security Parameters. See *Updating security parameters as manual clean step* for user guidance on usage. The supported security parameters for this clean step are: `Password_Complexity`, `RequiredLoginForiLORBSU`, `IPMI/DCMI_Over_LAN`, `RequireHostAuthentication` and `Secure_Boot`.

update_minimum_password_length: Updates the Minimum Password Length security parameter. See *Update Minimum Password Length security parameter as manual clean step* for user guidance on usage.

update_auth_failure_logging_threshold: Updates the Authentication Failure Logging security parameter. See *Update Authentication Failure Logging security parameter as manual clean step* for user guidance on usage.

- iLO with firmware version 1.5 is minimally required to support all the operations.

For more information on node manual cleaning, see *Manual cleaning*

Node Deployment Customization

The hardware type ilo and ilo5 supports customization of node deployment via deploy templates, see *Node Deployment*.

The supported deploy steps are:

- **apply_configuration:** Applies given BIOS settings on the node. See *BIOS configuration support*. This step is part of the bios interface.
- **factory_reset:** Resets the BIOS settings on the node to factory defaults. See *BIOS configuration support*. This step is part of the bios interface.
- **reset_bios_to_default:** Resets system ROM settings to default. This step is supported only on Gen9 and above servers. This step is part of the management interface.
- **reset_secure_boot_keys_to_default:** Resets secure boot keys to manufacturers defaults. This step is supported only on Gen9 and above servers. This step is part of the management interface.
- **reset_ilo_credential:** Resets the iLO password. The password need to be specified in ilo_password argument of the step. This step is part of the management interface.
- **clear_secure_boot_keys:** Clears all secure boot keys. This step is supported only on Gen9 and above servers. This step is part of the management interface.
- **reset_ilo:** Resets the iLO. This step is part of the management interface.
- **update_firmware:** Updates the firmware of the devices. This step is part of the management interface. See *Initiating firmware update as manual clean step* for user guidance on usage. The supported devices for firmware update are: ilo, cpld, power_pic, bios and chassis. This step is part of management interface. Please refer to below table for their commonly used descriptions.

Device	Description
ilo	BMC for HPE ProLiant servers
cpld	System programmable logic device
power_pic	Power management controller
bios	HPE ProLiant System ROM
chassis	System chassis device

Some devices firmware cannot be updated via this method, such as: storage controllers, host bus adapters, disk drive firmware, network interfaces and Onboard Administrator (OA).

- **flash_firmware_sum:** Updates all or list of user specified firmware components on the node using Smart Update Manager (SUM). It is an inband step associated with the management interface. See *Smart Update Manager (SUM) based firmware update* for more information on usage.

- **apply_configuration:** Applies RAID configuration on the node. See *RAID Configuration* for more information. This step is part of the `raid` interface.

Example of using deploy template with the Compute service

Create a deploy template with a single step:

```
baremetal deploy template create \
  CUSTOM_HYPERTHREADING_ON \
  --steps '[{"interface": "bios", "step": "apply_configuration", "args": {
↪ "settings": [{"name": "ProcHyperthreading", "value": "Enabled"}]}, "priority
↪ ": 150}]'
```

Add the trait `CUSTOM_HYPERTHREADING_ON` to the node represented by `$node_ident`:

```
baremetal node add trait $node_ident CUSTOM_HYPERTHREADING_ON
```

Update the flavor `bm-hyperthreading-on` in the Compute service with the following property:

```
openstack flavor set --property trait:CUSTOM_HYPERTHREADING_ON=required bm-
↪ hyperthreading-on
```

Creating a Compute instance with this flavor will ensure that the instance is scheduled only to Bare Metal nodes with the `CUSTOM_HYPERTHREADING_ON` trait. When an instance is created using the `bm-hyperthreading-on` flavor, then the deploy steps of deploy template `CUSTOM_HYPERTHREADING_ON` will be executed during the deployment of the scheduled node, causing Hyperthreading to be enabled in the nodes BIOS configuration.

Hardware Inspection Support

The hardware type `ilo` supports hardware inspection.

Note:

- The disk size is returned by RIBCL/RIS only when RAID is preconfigured on the storage. If the storage is Direct Attached Storage, then RIBCL/RIS fails to get the disk size.
- The SNMPv3 inspection gets disk size for all types of storages. If RIBCL/RIS is unable to get disk size and SNMPv3 inspection is requested, the `proliantutils` does SNMPv3 inspection to get the disk size. If `proliantutils` is unable to get the disk size, it raises an error. This feature is available in `proliantutils` release version `>= 2.2.0`.
- The iLO must be updated with SNMPv3 authentication details. Please refer to the section *SNMPv3 Authentication* in *HPE iLO4 User Guide* for setting up authentication details on iLO. The following parameters are mandatory to be given in `driver_info` for SNMPv3 inspection:
 - `snmp_auth_user` : The SNMPv3 user.
 - `snmp_auth_prot_password` : The auth protocol pass phrase.
 - `snmp_auth_priv_password` : The privacy protocol pass phrase.

The following parameters are optional for SNMPv3 inspection:

- `snmp_auth_protocol` : The Auth Protocol. The valid values are MD5 and SHA. The iLO default value is MD5.
 - `snmp_auth_priv_protocol` : The Privacy protocol. The valid values are AES and DES. The iLO default value is DES.
-

The inspection process will discover the following essential properties (properties required for scheduling deployment):

- `memory_mb`: memory size
- `cpus`: number of cpus
- `cpu_arch`: cpu architecture
- `local_gb`: disk size

Inspection can also discover the following extra capabilities for iLO driver:

- `ilo_firmware_version`: iLO firmware version
- `rom_firmware_version`: ROM firmware version
- `secure_boot`: secure boot is supported or not. The possible values are true or false. The value is returned as true if secure boot is supported by the server.
- `server_model`: server model
- `pci_gpu_devices`: number of gpu devices connected to the bare metal.
- `nic_capacity`: the max speed of the embedded NIC adapter.
- `sriov_enabled`: true, if server has the SRIOV supporting NIC.
- `has_rotational`: true, if server has HDD disk.
- `has_ssd`: true, if server has SSD disk.
- `has_nvme_ssd`: true, if server has NVME SSD disk.
- `cpu_vt`: true, if server supports cpu virtualization.
- `hardware_supports_raid`: true, if RAID can be configured on the server using RAID controller.
- `nvdimn_n`: true, if server has NVDIMM_N type of persistent memory.
- `persistent_memory`: true, if server has persistent memory.
- `logical_nvdimn_n`: true, if server has logical NVDIMM_N configured.
- `rotational_drive_<speed>_rpm`: The capabilities `rotational_drive_4800_rpm`, `rotational_drive_5400_rpm`, `rotational_drive_7200_rpm`, `rotational_drive_10000_rpm` and `rotational_drive_15000_rpm` are set to true if the server has HDD drives with speed of 4800, 5400, 7200, 10000 and 15000 rpm respectively.
- `logical_raid_level_<raid_level>`: The capabilities `logical_raid_level_0`, `logical_raid_level_1`, `logical_raid_level_2`, `logical_raid_level_5`, `logical_raid_level_6`, `logical_raid_level_10`, `logical_raid_level_50` and `logical_raid_level_60` are set to true if any of the raid levels among 0, 1, 2, 5, 6, 10, 50 and 60 are configured on the system.

- `overall_security_status`: `Ok` or `Risk` or `Ignored` as returned by iLO security dashboard. iLO computes the overall security status by evaluating the security status for each of the security parameters. Admin needs to fix the actual parameters and then re-inspect so that iLO can recompute the overall security status. If the all security params, whose `security_status` is `Risk`, have the `Ignore` field set to `True`, then iLO sets the overall security status value as `Ignored`. All the security params must have the `security_status` as `Ok` for the `overall_security_status` to have the value as `Ok`.
- `last_firmware_scan_status`: `Ok` or `Risk` as returned by iLO security dashboard. This denotes security status of the last firmware scan done on the system. If it is `Risk`, the recommendation is to run `clean_step update_firmware_sum` without any specific firmware components so that firmware is updated for all the components using latest SPP (Service Provider Pack) ISO and then re-inspect to get the security status again.
- `security_override_switch`: `Ok` or `Risk` as returned by iLO security dashboard. This is disable/enable login to the iLO using credentials. This can be toggled only by physical visit to the bare metal.
- `gpu_<vendor>_count`: Integer value. The capability name is dynamically formed as `gpu_<vendor>_count`. The vendor name is replaced in the `<vendor>`. If the vendor name is not returned by the hardware, then vendor ID in hexadecimal form is replaced in the capability name. Examples: `{gpu_Nvidia_count: 1}`, `{gpu_0x102b_count: 1}`.
- `gpu_<vendor_device_name>_count`: Integer value. The capability name is formed dynamically by replacing the gpu device name as returned by ilo in `<vendor_device_name>`. Examples: `{gpu_Nvidia_Tesla_M10_count: 1}`, `{gpu_Embedded_Video_Controller_count: 1}`
- `gpu_<vendor_device_name>`: Boolean. The capability name is formed dynamically by replacing the gpu device name as returned by ilo in `<vendor_device_name>`. Examples: `{gpu_Nvidia_Tesla_M10: True}`, `{gpu_Embedded_Video_Controller: True}`

Note:

- The capability `nic_capacity` can only be discovered if `ipmitool` version `>= 1.8.15` is used on the conductor. The latest version can be downloaded from [here](#).
- The iLO firmware version needs to be 2.10 or above for `nic_capacity` to be discovered.
- To discover IPMI based attributes you need to enable iLO feature IPMI/DCMI over LAN Access on `iLO4` and `iLO5` management engine.
- The `proliantutils` returns only active NICs for Gen10 ProLiant HPE servers. The user would need to delete the ironic ports corresponding to inactive NICs for Gen8 and Gen9 servers as `proliantutils` returns all the discovered (active and otherwise) NICs for Gen8 and Gen9 servers and ironic ports are created for all of them. Inspection logs a warning if the node under inspection is Gen8 or Gen9.
- The security dashboard capabilities are applicable only for Gen10 ProLiant HPE servers and above. To fix the security dashboard parameters value from `Risk` to `Ok`, user need to fix the parameters separately and re-inspect to see the security status of the parameters.

The operator can specify these capabilities in nova flavor for node to be selected for scheduling:

```
nova flavor-key my-baremetal-flavor set capabilities:server_model="<in> Gen8"
```

(continues on next page)

(continued from previous page)

```
nova flavor-key my-baremetal-flavor set capabilities:nic_capacity="10Gb"
nova flavor-key my-baremetal-flavor set capabilities:ilo_firmware_version="
↳<in> 2.10"
nova flavor-key my-baremetal-flavor set capabilities:has_ssd="true"
```

See *Capabilities discovery* for more details and examples.

Swiftless deploy for intermediate images

The hardware type `ilo` with `ilo-virtual-media` as boot interface can deploy and boot the server with and without `swift` being used for hosting the intermediate temporary floppy image (holding metadata for deploy kernel and ramdisk) and the boot ISO. A local HTTP(S) web server on each conductor node needs to be configured. Please refer to *Web server configuration on conductor* for more information. The HTTPS web server needs to be enabled (instead of HTTP web server) in order to send management information and images in encrypted channel over HTTPS.

Note: This feature assumes that the user inputs are on Glance which uses `swift` as backend. If `swift` dependency has to be eliminated, please refer to *HTTP(S) Based Deploy Support* also.

Deploy Process

Please refer to *Netboot in swiftless deploy for intermediate images* for partition image support and *Localboot in swiftless deploy for intermediate images* for whole disk image support.

HTTP(S) Based Deploy Support

The user input for the images given in `driver_info` like `ilo_deploy_iso`, `deploy_kernel` and `deploy_ramdisk` and in `instance_info` like `image_source`, `kernel`, `ramdisk` and `ilo_boot_iso` may also be given as HTTP(S) URLs.

The HTTP(S) web server can be configured in many ways. For the Apache web server on Ubuntu, refer [here](#). The web server may reside on a different system than the conductor nodes, but its URL must be reachable by the conductor and the bare metal nodes.

Deploy Process

Please refer to *Netboot with HTTP(S) based deploy* for partition image boot and *Localboot with HTTP(S) based deploy* for whole disk image boot.

Support for iLO driver with Standalone Ironic

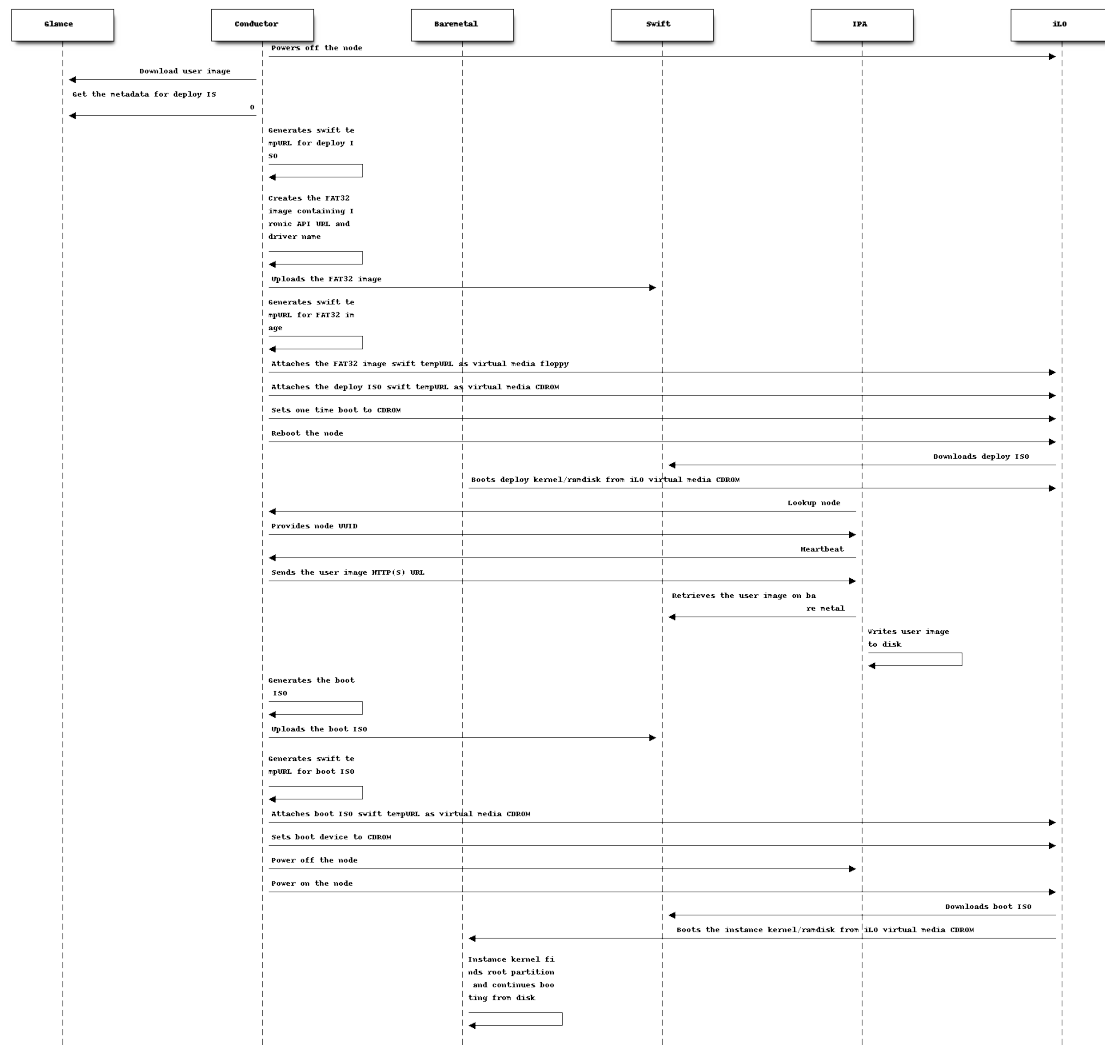
It is possible to use ironic as standalone services without other OpenStack services. The `ilo` hardware type can be used in standalone ironic. This feature is referred to as `ilo driver with standalone ironic` in this document.

Configuration

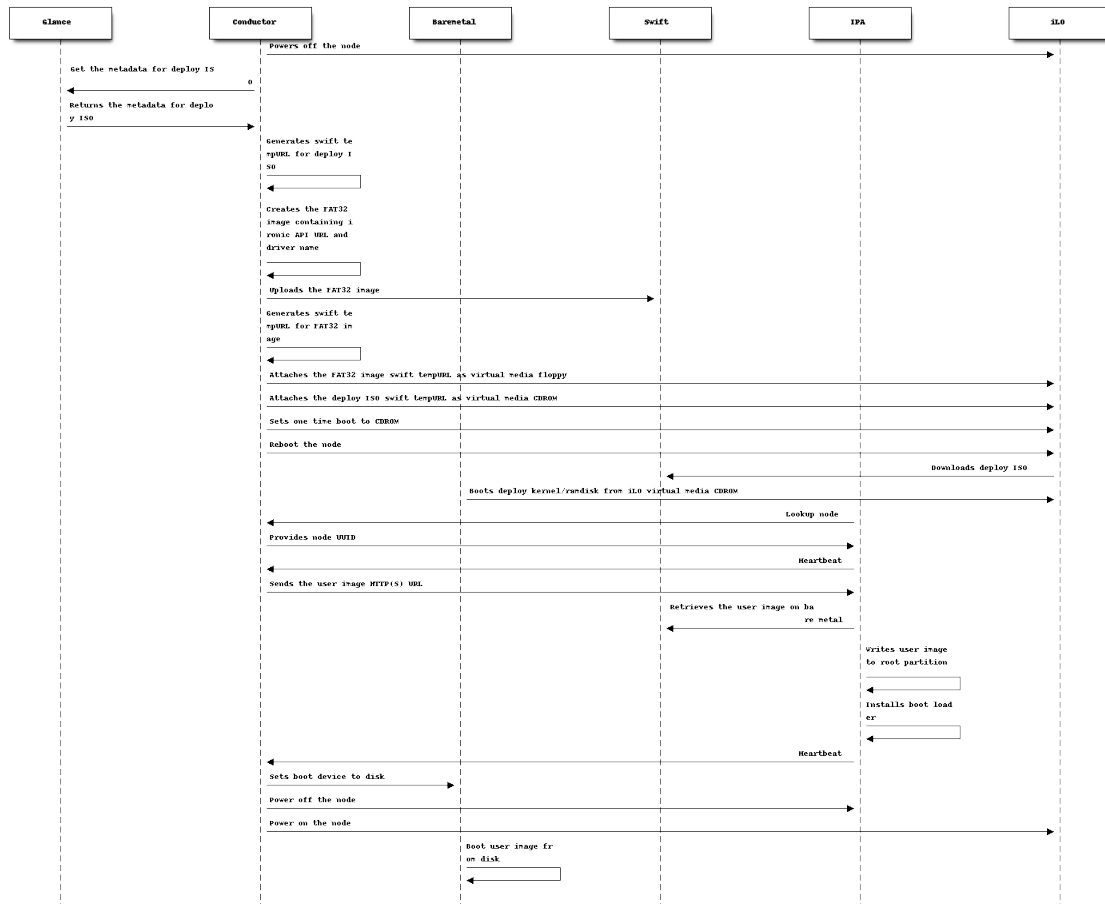
The HTTP(S) web server needs to be configured as described in *HTTP(S) Based Deploy Support* and *Web server configuration on conductor* needs to be configured for hosting intermediate images on conductor as described in *Swiftless deploy for intermediate images*.

Deploy Process

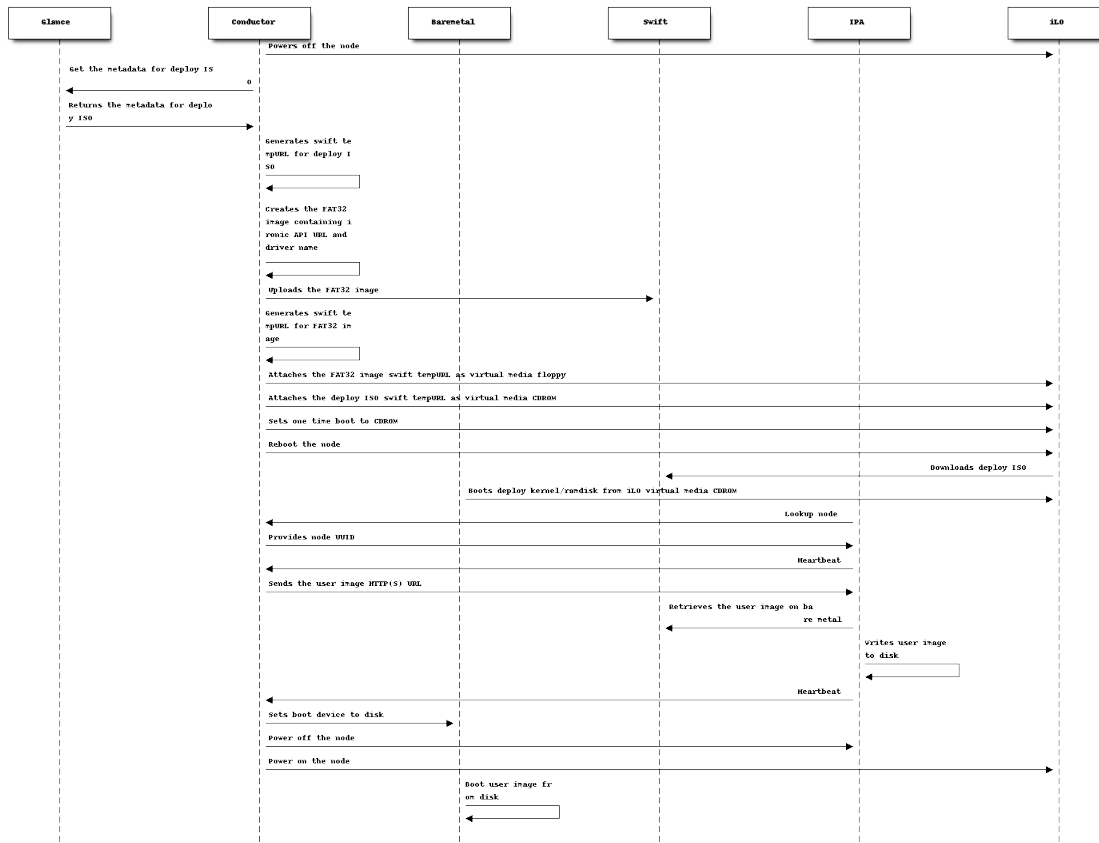
Netboot with glance and swift



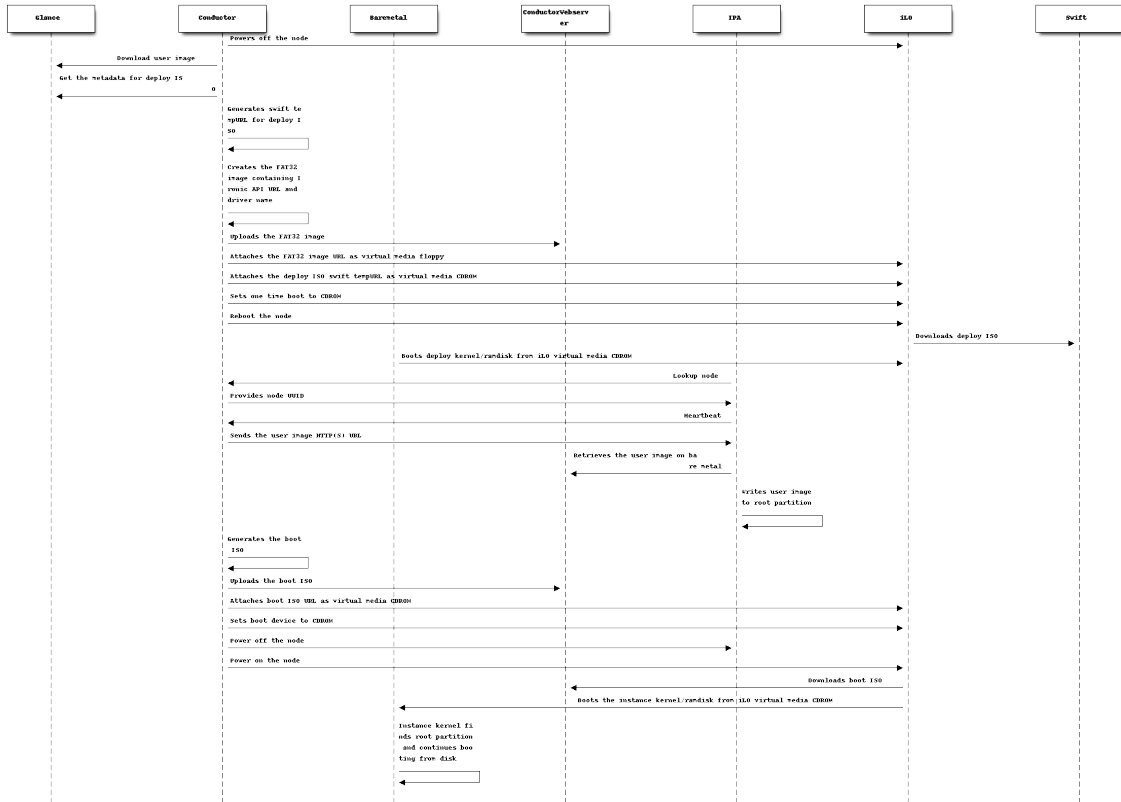
Localboot with glance and swift for partition images



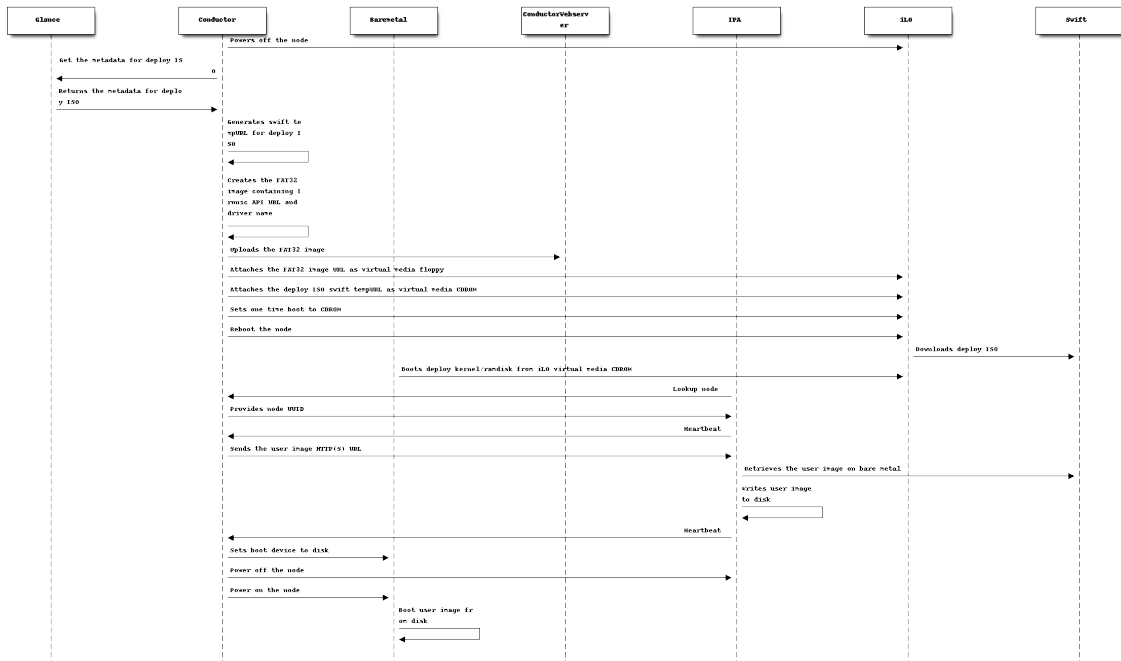
Localboot with glance and swift



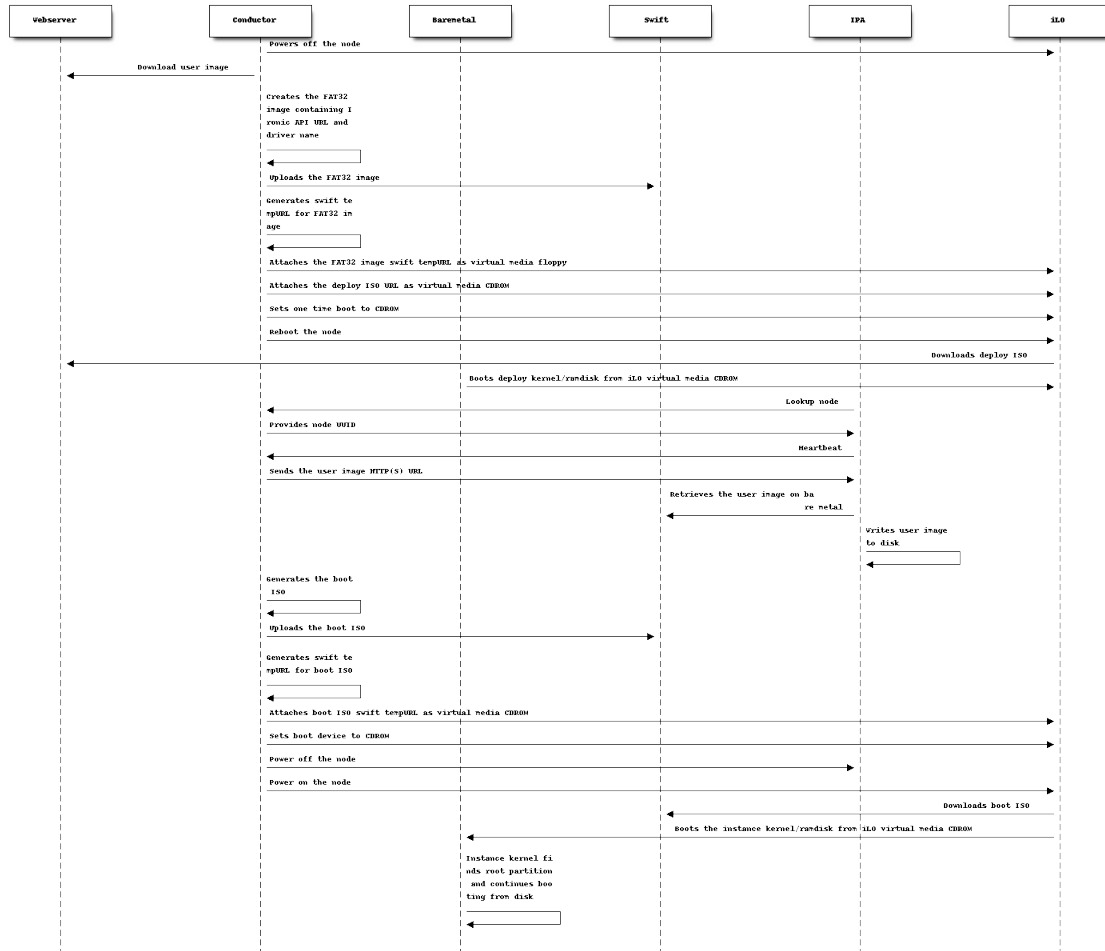
Netboot in swiftless deploy for intermediate images



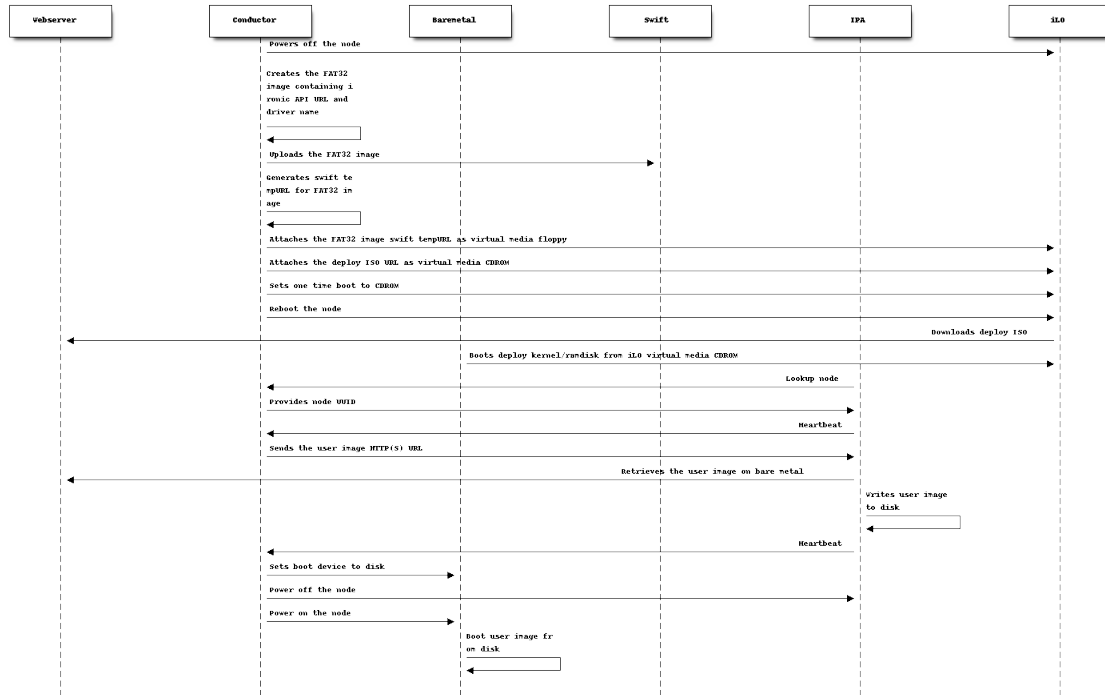
Localboot in swiftless deploy for intermediate images



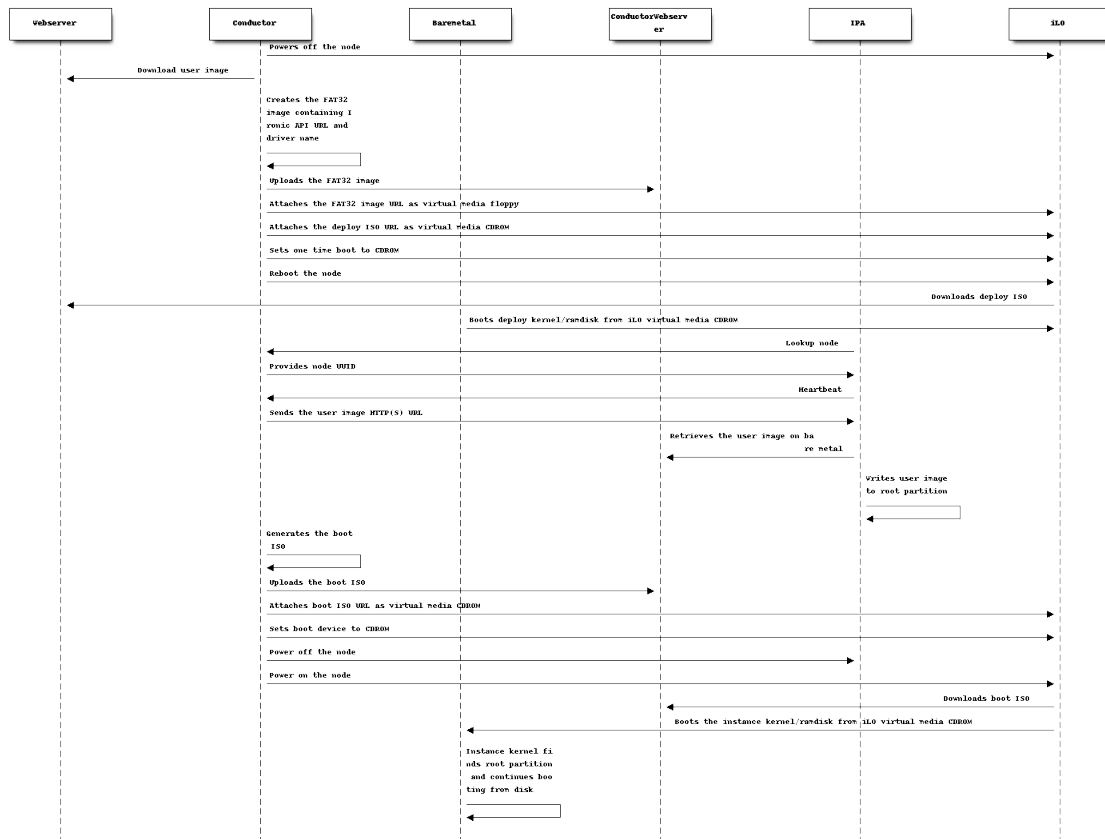
Netboot with HTTP(S) based deploy



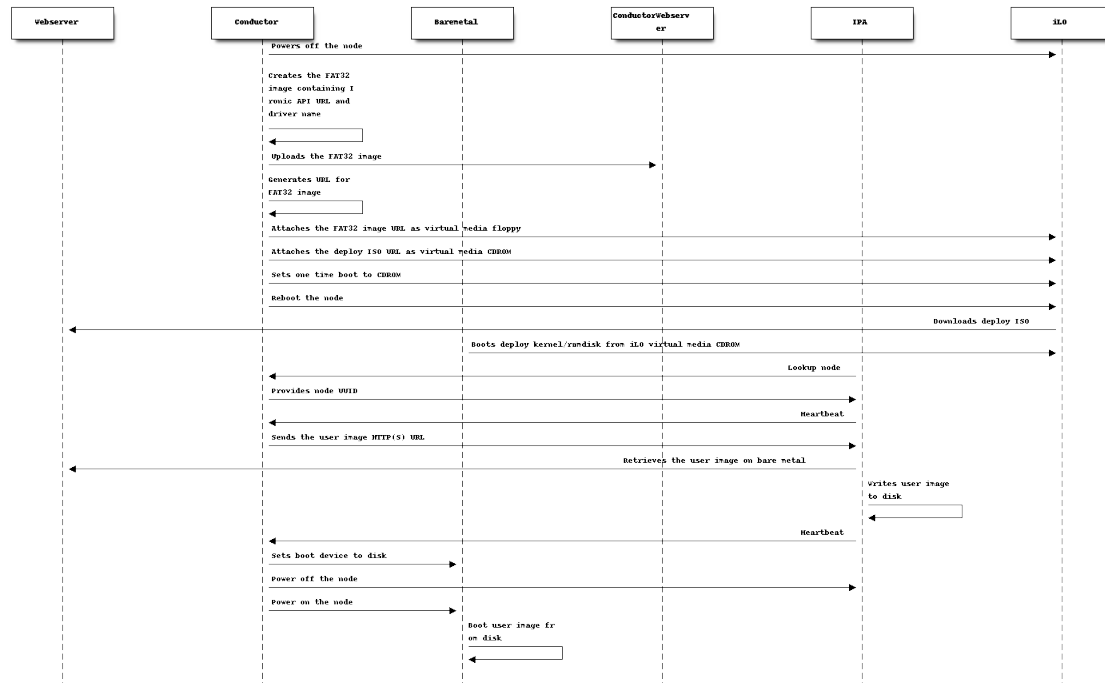
Localboot with HTTP(S) based deploy



Netboot in standalone ironic



Localboot in standalone ironic



Activating iLO Advanced license as manual clean step

iLO driver can activate the iLO Advanced license key as a manual cleaning step. Any manual cleaning step can only be initiated when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again. User can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node.

An example of a manual clean step with `activate_license` as the only clean step could be:

```

"clean_steps": [{
  "interface": "management",
  "step": "activate_license",
  "args": {
    "ilo_license_key": "ABC12-XXXXX-XXXXX-XXXXX-YZ345"
  }
}]
  
```

The different attributes of `activate_license` clean step are as follows:

Attribute	Description
<code>interface</code>	Interface of clean step, here <code>management</code>
<code>step</code>	Name of clean step, here <code>activate_license</code>
<code>args</code>	Keyword-argument entry (<code><name></code> : <code><value></code>) being passed to clean step
<code>args.ilo_license_key</code>	iLO Advanced license key to activate enterprise features. This is mandatory.

Initiating firmware update as manual clean step

iLO driver can invoke secure firmware update as a manual cleaning step. Any manual cleaning step can only be initiated when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again. A user can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node.

An example of a manual clean step with `update_firmware` as the only clean step could be:

```
"clean_steps": [{
  "interface": "management",
  "step": "update_firmware",
  "args": {
    "firmware_update_mode": "ilo",
    "firmware_images": [
      {
        "url": "file:///firmware_images/ilo/1.5/CP024444.scexe",
        "checksum": "a94e683ea16d9ae44768f0a65942234d",
        "component": "ilo"
      },
      {
        "url": "swift://firmware_container/cpld2.3.rpm",
        "checksum": "<md5-checksum-of-this-file>",
        "component": "cpld"
      },
      {
        "url": "http://my_address:port/firmwares/bios_vLatest.scexe",
        "checksum": "<md5-checksum-of-this-file>",
        "component": "bios"
      },
      {
        "url": "https://my_secure_address_url/firmwares/chassis_
↪vLatest.scexe",
        "checksum": "<md5-checksum-of-this-file>",
        "component": "chassis"
      },
      {
        "url": "file:///home/ubuntu/firmware_images/power_pic/pmc_v3.
↪0.bin",
        "checksum": "<md5-checksum-of-this-file>",
        "component": "power_pic"
      }
    ]
  }
}]
```

The different attributes of `update_firmware` clean step are as follows:

Attribute	Description
interface	Interface of clean step, here management
step	Name of clean step, here update_firmware
args	Keyword-argument entry (<name>: <value>) being passed to clean step
args. firmware_update_mode	Mode (or mechanism) of out-of-band firmware update. Supported value is ilo. This is mandatory.
args. firmware_images	Ordered list of dictionaries of images to be flashed. This is mandatory.

Each firmware image block is represented by a dictionary (JSON), in the form:

```
{
  "url": "<url of firmware image file>",
  "checksum": "<md5 checksum of firmware image file to verify the image>",
  "component": "<device on which firmware image will be flashed>"
}
```

All the fields in the firmware image block are mandatory.

- The different types of firmware url schemes supported are: `file`, `http`, `https` and `swift`.

Note: This feature assumes that while using `file` url scheme the file path is on the conductor controlling the node.

Note: The `swift` url scheme assumes the `swift` account of the `service` project. The `service` project (tenant) is a special project created in the Keystone system designed for the use of the core OpenStack services. When Ironic makes use of Swift for storage purpose, the account is generally `service` and the container is generally `ironic` and `ilo` driver uses a container named `ironic_ilo_container` for their own purpose.

Note: While using firmware files with a `.rpm` extension, make sure the commands `rpm2cpio` and `cpio` are present on the conductor, as they are utilized to extract the firmware image from the package.

- The firmware components that can be updated are: `ilo`, `cpld`, `power_pic`, `bios` and `chassis`.
- The firmware images will be updated in the order given by the operator. If there is any error during processing of any of the given firmware images provided in the list, none of the firmware updates will occur. The processing error could happen during image download, image checksum verification or image extraction. The logic is to process each of the firmware files and update them on the devices only if all the files are processed successfully. If, during the update (uploading and flashing) process, an update fails, then the remaining updates, if any, in the list will be aborted. But it is recommended to triage and fix the failure and re-attempt the manual clean step `update_firmware` for the aborted `firmware_images`.

The devices for which the firmwares have been updated successfully would start functioning using their newly updated firmware.

- As a troubleshooting guidance on the complete process, check Ironic conductor logs carefully to see if there are any firmware processing or update related errors which may help in root causing or gain an understanding of where things were left off or where things failed. You can then fix or work around and then try again. A common cause of update failure is HPE Secure Digital Signature check failure for the firmware image file.
- To compute md5 checksum for your image file, you can use the following command:

```
$ md5sum image.rpm
66cdb090c80b71daa21a67f06ecd3f33  image.rpm
```

Smart Update Manager (SUM) based firmware update

The firmware update based on **SUM** is an inband clean/deploy step supported by iLO driver. The firmware update is performed on all or list of user specified firmware components on the node. Refer to [SUM User Guide](#) to get more information on SUM based firmware update.

Note: `update_firmware_sum` clean step requires the agent ramdisk with Proliant Hardware Manager from the `proliantutils` version 2.5.0 or higher. See *DIB support for Proliant Hardware Manager* to create the agent ramdisk with Proliant Hardware Manager.

Note: `flash_firmware_sum` deploy step requires the agent ramdisk with Proliant Hardware Manager from the `proliantutils` version 2.9.5 or higher. See *DIB support for Proliant Hardware Manager* to create the agent ramdisk with Proliant Hardware Manager.

The attributes of `update_firmware_sum/flash_firmware_sum` step are as follows:

Attribute	Description
<code>interface</code>	Interface of the clean step, here management
<code>step</code>	Name of the clean step, here <code>update_firmware_sum</code>
<code>args</code>	Keyword-argument entry (<name>: <value>) being passed to the clean step

The keyword arguments used for the step are as follows:

- `url`: URL of SPP (Service Pack for Proliant) ISO. It is mandatory. The URL schemes supported are `http`, `https` and `swift`.
- `checksum`: MD5 checksum of SPP ISO to verify the image. It is mandatory.
- `components`: List of filenames of the firmware components to be flashed. It is optional. If not provided, the firmware update is performed on all the firmware components.

The step performs an update on all or a list of firmware components and returns the SUM log files. The log files include `hpsum_log.txt` and `hpsum_detail_log.txt` which holds the information about firmware components, firmware version for each component and their update status. The log object will be named with the following pattern:

```
<node>[_<instance-uuid>]_update_firmware_sum_<timestamp yyyy-mm-dd-hh-mm-ss>.
↳tar.gz
or
<node>[_<instance-uuid>]_flash_firmware_sum_<timestamp yyyy-mm-dd-hh-mm-ss>.
↳tar.gz
```

Refer to *Retrieving logs from the deploy ramdisk* for more information on enabling and viewing the logs returned from the ramdisk.

An example of `update_firmware_sum` clean step:

```
{
  "interface": "management",
  "step": "update_firmware_sum",
  "args":
    {
      "url": "http://my_address:port/SPP.iso",
      "checksum": "abcdefxyz",
      "components": ["CP024356.scexe", "CP008097.exe"]
    }
}
```

The step fails if there is any error in the processing of step arguments. The processing error could happen during validation of components file extension, image download, image checksum verification or image extraction. In case of a failure, check Ironic conductor logs carefully to see if there are any validation or firmware processing related errors which may help in root cause analysis or gaining an understanding of where things were left off or where things failed. You can then fix or work around and then try again.

Warning: This feature is officially supported only with RHEL and SUSE based IPA ramdisk. Refer to [SUM](#) for supported OS versions for specific SUM version.

Note: Refer [Guidelines for SPP ISO](#) for steps to get SPP (Service Pack for ProLiant) ISO.

Updating security parameters as manual clean step

iLO driver can invoke security parameters update as a manual clean step. Any manual cleaning step can only be initiated when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again. A user can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node. This feature is only supported for iLO5 based hardware.

An example of a manual clean step with `security_parameters_update` as the only clean step could be:

```
"clean_steps": [{
  "interface": "management",
  "step": "security_parameters_update",
  "args": {
```

(continues on next page)

(continued from previous page)

```

"security_parameters":[
  {
    "param": "password_complexity",
    "enable": "True",
    "ignore": "False"
  },
  {
    "param": "require_login_for_ilo_rbsu",
    "enable": "True",
    "ignore": "False"
  },
  {
    "param": "ipmi_over_lan",
    "enable": "True",
    "ignore": "False"
  },
  {
    "param": "secure_boot",
    "enable": "True",
    "ignore": "False"
  },
  {
    "param": "require_host_authentication",
    "enable": "True",
    "ignore": "False"
  }
]
}
}]

```

The different attributes of `security_parameters_update` clean step are as follows:

Attribute	Description
<code>interface</code>	Interface of clean step, here management
<code>step</code>	Name of clean step, here <code>security_parameters_update</code>
<code>args</code>	Keyword-argument entry (<name>: <value>) being passed to clean step
<code>args.security_parameters</code>	Ordered list of dictionaries of security parameters to be updated. This is mandatory.

Each security parameter block is represented by a dictionary (JSON), in the form:

```

{
  "param": "<security parameter name>",
  "enable": "security parameter to be enabled/disabled",
  "ignore": "security parameter status to be ignored or not"
}

```

In all of these fields, `param` field is mandatory. Remaining fields are boolean and are optional. If user doesn't pass any value then for `enable` field the default will be `True` and for `ignore` field default will be `False`.

- The Security Parameters which are supported for this clean step are: Password_Complexity, RequiredLoginForiLORBSU, RequireHostAuthentication, IPMI/DCMI_Over_LAN and Secure_Boot.

Update Minimum Password Length security parameter as manual clean step

iLO driver can invoke Minimum Password Length security parameter update as a manual clean step. This feature is only supported for iLO5 based hardware.

An example of a manual clean step with `update_minimum_password_length` as the only clean step could be:

```
"clean_steps": [{
  "interface": "management",
  "step": "update_minimum_password_length",
  "args": {
    "password_length": "8",
    "ignore": "False"
  }
}]
```

Both the arguments `password_length` and `ignore` are optional. The accepted values for `password_length` are 0 to 39. If user doesn't pass any value, the default value for `password_length` will be 8 and for `ignore` the default value be False.

Update Authentication Failure Logging security parameter as manual clean step

iLO driver can invoke Authentication Failure Logging security parameter update as a manual clean step. This feature is only supported for iLO5 based hardware.

An example of a manual clean step with Authentication Failure Logging as the only clean step could be:

```
"clean_steps": [{
  "interface": "management",
  "step": "update_auth_failure_logging_threshold",
  "args": {
    "logging_threshold": "1",
    "ignore": "False"
  }
}]
```

Both the arguments `logging_threshold` and `ignore` are optional. The accepted values for `logging_threshold` are 0 to 5. If user doesn't pass any value, the default value for `logging_threshold` will be 1 and for `ignore` the default value be False. If user passes the value of `logging_threshold` as 0, the Authentication Failure Logging security parameter will be disabled.

RAID Support

The inband RAID functionality is supported by iLO driver. See *RAID Configuration* for more information. Bare Metal service update node with following information after successful configuration of RAID:

- Node `properties/local_gb` is set to the size of root volume.
- Node `properties/root_device` is filled with `wwn` details of root volume. It is used by iLO driver as root device hint during provisioning.
- The value of raid level of root volume is added as `raid_level` capability to the nodes `capabilities` parameter within `properties` field. The operator can specify the `raid_level` capability in nova flavor for node to be selected for scheduling:

```
nova flavor-key ironic-test set capabilities:raid_level="1+0"  
nova boot --flavor ironic-test --image test-image instance-1
```

DIB support for Proliant Hardware Manager

Install `ironic-python-agent-builder` following the guide¹

To create an agent ramdisk with Proliant Hardware Manager, use the `proliant-tools` element in DIB:

```
ironic-python-agent-builder -o proliant-agent-ramdisk -e proliant-tools fedora
```

Disk Erase Support

`erase_devices` is an inband clean step supported by iLO driver. It performs erase on all the disks including the disks visible to OS as well as the raw disks visible to the Smart Storage Administrator (SSA).

This inband clean step requires `ssacli` utility starting from version 2.60-19.0 to perform the erase on physical disks. See the [ssacli documentation](#) for more information on `ssacli` utility and different erase methods supported by SSA.

The disk erasure via `shred` is used to erase disks visible to the OS and its implementation is available in Ironic Python Agent. The raw disks connected to the Smart Storage Controller are erased using Sanitize erase which is a `ssacli` supported erase method. If Sanitize erase is not supported on the Smart Storage Controller the disks are erased using One-pass erase (overwrite with zeros).

This clean step is supported when the agent ramdisk contains the Proliant Hardware Manager from the `proliantutils` version 2.3.0 or higher. This clean step is performed as part of automated cleaning and it is disabled by default. See *In-band vs out-of-band* for more information on enabling/disabling a clean step.

Install `ironic-python-agent-builder` following the guide¹

To create an agent ramdisk with Proliant Hardware Manager, use the `proliant-tools` element in DIB:

¹ *ironic-python-agent-builder*: <https://docs.openstack.org/ironic-python-agent-builder/latest/install/index.html>

```
ironic-python-agent-builder -o proliant-agent-ramdisk -e proliant-tools fedora
```

See the `proliant-tools` for more information on creating agent ramdisk with `proliant-tools` element in DIB.

Firmware based UEFI iSCSI boot from volume support

With Gen9 (UEFI firmware version 1.40 or higher) and Gen10 HPE Proliant servers, the driver supports firmware based UEFI boot of an iSCSI cinder volume.

This feature requires the node to be configured to boot in UEFI boot mode, as well as user image should be UEFI bootable image, and `PortFast` needs to be enabled in switch configuration for immediate spanning tree forwarding state so it wouldnt take much time setting the iSCSI target as persistent device.

The driver does not support this functionality when in bios boot mode. In case the node is configured with `ilo-pxe` or `ilo-ipxe` as boot interface and the boot mode configured on the bare metal is bios, the iscsi boot from volume is performed using iPXE. See *Boot From Volume* for more details.

To use this feature, configure the boot mode of the bare metal to `uefi` and configure the corresponding ironic node using the steps given in *Boot From Volume*. In a cloud environment with nodes configured to boot from bios and uefi boot modes, the virtual media driver only supports uefi boot mode, and that attempting to use iscsi boot at the same time with a bios volume will result in an error.

BIOS configuration support

The `ilo` and `ilo5` hardware types support `ilo` BIOS interface. The support includes providing manual clean steps *apply_configuration* and *factory_reset* to manage supported BIOS settings on the node. See *BIOS Configuration* for more details and examples.

Note: Prior to the Stein release the user is required to reboot the node manually in order for the settings to take into effect. Starting with the Stein release, iLO drivers reboot the node after running clean steps related to the BIOS configuration. The BIOS settings are cached and the clean step is marked as success only if all the requested settings are applied without any failure. If application of any of the settings fails, the clean step is marked as failed and the settings are not cached.

Configuration

Following are the supported BIOS settings and the corresponding brief description for each of the settings. For a detailed description please refer to [HPE Integrated Lights-Out REST API Documentation](#).

- **AdvancedMemProtection:** Configure additional memory protection with ECC (Error Checking and Correcting). Allowed values are `AdvancedEcc`, `OnlineSpareAdvancedEcc`, `MirroredAdvancedEcc`.
- **AutoPowerOn:** Configure the server to automatically power on when AC power is applied to the system. Allowed values are `AlwaysPowerOn`, `AlwaysPowerOff`, `RestoreLastState`.
- **BootMode:** Select the boot mode of the system. Allowed values are `Uefi`, `LegacyBios`

- **BootOrderPolicy:** Configure how the system attempts to boot devices per the Boot Order when no bootable device is found. Allowed values are `RetryIndefinitely`, `AttemptOnce`, `ResetAfterFailed`.
- **CollabPowerControl:** Enables the Operating System to request processor frequency changes even if the Power Regulator option on the server configured for Dynamic Power Savings Mode. Allowed values are `Enabled`, `Disabled`.
- **DynamicPowerCapping:** Configure when the System ROM executes power calibration during the boot process. Allowed values are `Enabled`, `Disabled`, `Auto`.
- **DynamicPowerResponse:** Enable the System BIOS to control processor performance and power states depending on the processor workload. Allowed values are `Fast`, `Slow`.
- **IntelligentProvisioning:** Enable or disable the Intelligent Provisioning functionality. Allowed values are `Enabled`, `Disabled`.
- **IntelPerfMonitoring:** Exposes certain chipset devices that can be used with the Intel Performance Monitoring Toolkit. Allowed values are `Enabled`, `Disabled`.
- **IntelProcVtd:** Hypervisor or operating system supporting this option can use hardware capabilities provided by Intels Virtualization Technology for Directed I/O. Allowed values are `Enabled`, `Disabled`.
- **IntelQpiFreq:** Set the QPI Link frequency to a lower speed. Allowed values are `Auto`, `MinQpiSpeed`.
- **IntelTxt:** Option to modify Intel TXT support. Allowed values are `Enabled`, `Disabled`.
- **PowerProfile:** Set the power profile to be used. Allowed values are `BalancedPowerPerf`, `MinPower`, `MaxPerf`, `Custom`.
- **PowerRegulator:** Determines how to regulate the power consumption. Allowed values are `DynamicPowerSavings`, `StaticLowPower`, `StaticHighPerf`, `OsControl`.
- **ProcAes:** Enable or disable the Advanced Encryption Standard Instruction Set (AES-NI) in the processor. Allowed values are `Enabled`, `Disabled`.
- **ProcCoreDisable:** Disable processor cores using Intels Core Multi-Processing (CMP) Technology. Allowed values are Integers ranging from 0 to 24.
- **ProcHyperthreading:** Enable or disable Intel Hyperthreading. Allowed values are `Enabled`, `Disabled`.
- **ProcNoExecute:** Protect your system against malicious code and viruses. Allowed values are `Enabled`, `Disabled`.
- **ProcTurbo:** Enables the processor to transition to a higher frequency than the processors rated speed using Turbo Boost Technology if the processor has available power and is within temperature specifications. Allowed values are `Enabled`, `Disabled`.
- **ProcVirtualization:** Enables or Disables a hypervisor or operating system supporting this option to use hardware capabilities provided by Intels Virtualization Technology. Allowed values are `Enabled`, `Disabled`.
- **SecureBootStatus:** The current state of Secure Boot configuration. Allowed values are `Enabled`, `Disabled`.

Note: This setting is read-only and cant be modified with `apply_configuration` clean step.

- **Sriov:** If enabled, SR-IOV support enables a hypervisor to create virtual instances of a PCI-express device, potentially increasing performance. If enabled, the BIOS allocates additional resources to PCI-express devices. Allowed values are `Enabled`, `Disabled`.
- **ThermalConfig:** select the fan cooling solution for the system. Allowed values are `OptimalCooling`, `IncreasedCooling`, `MaxCooling`
- **ThermalShutdown:** Control the reaction of the system to caution level thermal events. Allowed values are `Enabled`, `Disabled`.
- **TpmState:** Current TPM device state. Allowed values are `NotPresent`, `PresentDisabled`, `PresentEnabled`.

Note: This setting is read-only and cant be modified with `apply_configuration` clean step.

- **TpmType:** Current TPM device type. Allowed values are `NoTpm`, `Tpm12`, `Tpm20`, `Tm10`.

Note: This setting is read-only and cant be modified with `apply_configuration` clean step.

- **UefiOptimizedBoot:** Enables or Disables the System BIOS boot using native UEFI graphics drivers. Allowed values are `Enabled`, `Disabled`.
- **WorkloadProfile:** Change the Workload Profile to accomodate your desired workload. Allowed values are `GeneralPowerEfficientCompute`, `GeneralPeakFrequencyCompute`, `GeneralThroughputCompute`, `Virtualization-PowerEfficient`, `Virtualization-MaxPerformance`, `LowLatency`, `MissionCritical`, `TransactionalApplicationProcessing`, `HighPerformanceCompute`, `DecisionSupport`, `GraphicProcessing`, `I/OThroughput`, `Custom`

Note: This setting is only applicable to ProLiant Gen10 servers with iLO 5 management systems.

Certificate based validation in iLO

The driver supports validation of certificates on the HPE Proliant servers. The path to certificate file needs to be appropriately set in `ca_file` in the nodes `driver_info`. To update SSL certificates into iLO, refer to [HPE Integrated Lights-Out Security Technology Brief](#). Use iLO hostname or IP address as a Common Name (CN) while generating Certificate Signing Request (CSR). Use the same value as `ilo_address` while enrolling node to Bare Metal service to avoid SSL certificate validation errors related to hostname mismatch.

Rescue mode support

The hardware type `ilo` supports rescue functionality. Rescue operation can be used to boot nodes into a rescue ramdisk so that the `rescue` user can access the node.

Please refer to *Rescue Mode* for detailed explanation of rescue feature.

Inject NMI support

The management interface `ilo` supports injection of non-maskable interrupt (NMI) to a bare metal. Following command can be used to inject NMI on a server:

```
baremetal node inject nmi <node>
```

Following command can be used to inject NMI via Compute service:

```
openstack server dump create <server>
```

Note: This feature is supported on HPE ProLiant Gen9 servers and beyond.

Soft power operation support

The power interface `ilo` supports soft power off and soft reboot operations on a bare metal. Following commands can be used to perform soft power operations on a server:

```
baremetal node reboot --soft \  
  [--power-timeout <power-timeout>] <node>  
  
baremetal node power off --soft \  
  [--power-timeout <power-timeout>] <node>
```

Note: The configuration `[conductor]soft_power_off_timeout` is used as a default timeout value when no timeout is provided while invoking hard or soft power operations.

Note: Server POST state is used to track the power status of HPE ProLiant Gen9 servers and beyond.

Out of Band RAID Support

With Gen10 HPE Proliant servers and later the ilo5 hardware type supports firmware based RAID configuration as a clean step. This feature requires the node to be configured to ilo5 hardware type and its raid interface to be ilo5. See *RAID Configuration* for more information.

After a successful RAID configuration, the Bare Metal service will update the node with the following information:

- Node `properties/local_gb` is set to the size of root volume.
- Node `properties/root_device` is filled with `wwn` details of root volume. It is used by iLO driver as root device hint during provisioning.

Later the value of raid level of root volume can be added in `baremetal-with-RAID10` (RAID10 for raid level 10) resource class. And consequently flavor needs to be updated to request the resource class to create the server using selected node:

```
baremetal node set test_node --resource-class \  
baremetal-with-RAID10  
  
openstack flavor set --property \  
resources:CUSTOM_BAREMETAL_WITH_RAID10=1 test-flavor  
  
openstack server create --flavor test-flavor --image test-image instance-1
```

Note: Supported raid levels for ilo5 hardware type are: 0, 1, 5, 6, 10, 50, 60

IPv6 support

With the IPv6 support in `proliantutils` $\geq 2.8.0$, nodes can be enrolled into the baremetal service using the iLO IPv6 addresses.

```
baremetal node create --driver ilo --deploy-interface direct \  
--driver-info ilo_address=2001:0db8:85a3:0000:0000:8a2e:0370:7334 \  
--driver-info ilo_username=test-user \  
--driver-info ilo_password=test-password \  
--driver-info ilo_deploy_iso=test-iso \  
--driver-info ilo_rescue_iso=test-iso
```

Note: No configuration changes (in e.g. `ironic.conf`) are required in order to support IPv6.

Out of Band Sanitize Disk Erase Support

With Gen10 HPE Proliant servers and later the iLO5 hardware type supports firmware based sanitize disk erase as a clean step. This feature requires the node to be configured to iLO5 hardware type and its management interface to be iLO5.

The possible erase pattern its supports are:

- For HDD - overwrite, zero, crypto
- For SSD - block, zero, crypto

The default erase pattern are, for HDD, overwrite and for SSD, block.

Note: In average 300GB HDD with default pattern overwrite would take approx. 9 hours and 300GB SSD with default pattern block would take approx. 30 seconds to complete the erase.

Out of Band One Button Secure Erase Support

With Gen10 HPE Proliant servers which have been updated with SPP version 2019.03.0 or later the iLO5 hardware type supports firmware based one button secure erase as a clean step.

The One Button Secure Erase resets iLO and deletes all licenses stored there, resets BIOS settings, and deletes all Active Health System (AHS) and warranty data stored on the system. It also erases supported non-volatile storage data and deletes any deployment settings profiles. See [HPE Gen10 Security Reference Guide](#) for more information.

Below are the steps to perform this clean step:

- Perform the cleaning using `one_button_secure_erase` clean step

```
baremetal node clean $node_id --clean-steps\  
  '[{"interface": "management", "step": "one_button_secure_erase"}]'
```

- Once the clean step would triggered and node go to clean wait state and maintenance flag on node would be set to True, then delete the node

```
baremetal node delete $node_id
```

Note:

- Even after deleting the node, One Button Secure Erase operation would continue on the node.
- This clean step should be kept last if the multiple clean steps are to be executed. No clean step after this step would be executed.
- One Button Secure Erase should be used with extreme caution, and only when a system is being decommissioned. During the erase the iLO network would keep disconnecting and after the erase user will completely lose iLO access along with the credentials of the server, which needs to be regained by the administrator. The process can take up to a day or two to fully erase and reset all user data.
- When you activate One Button Secure Erase, iLO 5 does not allow firmware update or reset operations.

Note: Do not perform any iLO 5 configuration changes until this process is completed.

UEFI-HTTPS Boot support

The UEFI firmware on Gen10 HPE ProLiant servers supports booting from secured URLs. With this capability ilo5 hardware with ilo-uefi-https boot interface supports deploy/rescue features in more secured environments.

If swift is used as glance backend and ironic is configured to use swift to store temporary images, it is required that swift is configured on HTTPS so that the tempurl generated is HTTPS URL.

If the webserver is used for hosting the temporary images, then the webserver is required to serve requests on HTTPS.

If the images are hosted on a HTTPS webserver or swift configured with HTTPS with custom certificates, the user is required to export SSL certificates into iLO. Refer to [HPE Integrated Lights-Out Security Technology Brief](#) for more information.

The following command can be used to enroll a ProLiant node with ilo5 hardware type and ilo-uefi-https boot interface:

```
baremetal node create \  
  --driver ilo5 \  
  --boot-interface ilo-uefi-https \  
  --deploy-interface direct \  
  --raid-interface ilo5 \  
  --rescue-interface agent \  
  --driver-info ilo_address=<ilo-ip-address> \  
  --driver-info ilo_username=<ilo-username> \  
  --driver-info ilo_password=<ilo-password> \  
  --driver-info ilo_deploy_kernel=<glance-uuid-of-deploy-kernel> \  
  --driver-info ilo_deploy_ramdisk=<glance-uuid-of-rescue-ramdisk> \  
  --driver-info ilo_bootloader=<glance-uuid-of-bootloader>
```

Layer 3 or DHCP-less ramdisk booting

DHCP-less deploy is supported by `ilo` and `ilo5` hardware types. However it would work only with `ilo-virtual-media` boot interface. See *Layer 3 or DHCP-less ramdisk booting* for more information.

Intel IPMI driver

Overview

The `intel-ipmi` hardware type is same as the *IPMI driver* hardware type except for the support of Intel Speed Select Performance Profile (Intel *SST-PP*) feature. Intel SST-PP allows a server to run different workloads by configuring the CPU to run at 3 distinct operating points or profiles.

Intel SST-PP supports three configuration levels:

- 0 - Intel SST-PP Base Config
- 1 - Intel SST-PP Config 1
- 2 - Intel SST-PP Config 2

The following table shows the list of active cores and their base frequency at different SST-PP config levels:

Config	Cores	Base Freq (GHz)
Base	24	2.4
Config 1	20	2.5
Config 2	16	2.7

This configuration is managed by the management interface `intel-ipmitool` for IntelIPMI hardware.

IntelIPMI manages nodes by using *IPMI* (Intelligent Platform Management Interface) protocol versions 2.0 or 1.5. It uses the *IPMITool* utility which is an open-source command-line interface (CLI) for controlling IPMI-enabled devices.

Glossary

- IPMI - Intelligent Platform Management Interface.
- Intel SST-PP - Intel Speed Select Performance Profile.

Enabling the IntelIPMI hardware type

Please see *Configuring IPMI support* for the required dependencies.

1. To enable `intel-ipmi` hardware, add the following configuration to your `ironic.conf`:

```
[DEFAULT]
enabled_hardware_types = intel-ipmi
enabled_management_interfaces = intel-ipmitool
```

2. Restart the Ironic conductor service:

```
sudo service ironic-conductor restart

# Or, for RDO:
sudo systemctl restart openstack-ironic-conductor
```

Registering a node with the IntelIPMI driver

Nodes configured to use the IntelIPMI drivers should have the driver field set to `intel-ipmi`.

All the configuration value required for IntelIPMI is the same as the IPMI hardware type except the management interface which is `intel-ipmitool`. Refer *IPMI driver* for details.

The `baremetal node create` command can be used to enroll a node with an IntelIPMI driver. For example:

```
baremetal node create --driver intel-ipmi \
  --driver-info ipmi_address=<address> \
  --driver-info ipmi_username=<username> \
  --driver-info ipmi_password=<password>
```

Features of the intel-ipmi hardware type

Intel SST-PP

A node with Intel SST-PP can be configured to use it via `configure_intel_speedselect` deploy step. This deploy accepts:

- `intel_speedselect_config`: Hexadecimal code of Intel SST-PP configuration. Accepted values are 0x00, 0x01, 0x02. These values correspond to *Intel SST-PP Config Base*, *Intel SST-PP Config 1*, *Intel SST-PP Config 2* respectively. The input value must be a string.
- `socket_count`: Number of sockets in the node. The input value must be a positive integer (1 by default).

The deploy step issues an IPMI command with the raw code for each socket in the node to set the requested configuration. A reboot is required to reflect the changes.

Each configuration profile is mapped to traits that Ironic understands. Please note that these names are used for example purpose only. Any name can be used. Only the parameter value should match the deploy step `configure_intel_speedselect`.

- 0 - CUSTOM_INTEL_SPEED_SELECT_CONFIG_BASE
- 1 - CUSTOM_INTEL_SPEED_SELECT_CONFIG_1
- 2 - CUSTOM_INTEL_SPEED_SELECT_CONFIG_2

Now to configure a node with Intel SST-PP while provisioning, create deploy templates for each profiles in Ironic.

```
baremetal deploy template create \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_BASE \
```

(continues on next page)

(continued from previous page)

```
--steps '[{"interface": "management", "step": "configure_intel_speedselect
↪", "args": {"intel_speedselect_config": "0x00", "socket_count": 2},
↪"priority": 150}]'

baremetal deploy template create \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_1 \
  --steps '[{"interface": "management", "step": "configure_intel_speedselect
↪", "args": {"intel_speedselect_config": "0x01", "socket_count": 2},
↪"priority": 150}]'

baremetal deploy template create \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_2 \
  --steps '[{"interface": "management", "step": "configure_intel_speedselect
↪", "args": {"intel_speedselect_config": "0x02", "socket_count": 2},
↪"priority": 150}]'
```

All Intel SST-PP capable nodes should have these traits associated.

```
baremetal node add trait node-0 \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_BASE \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_1 \
  CUSTOM_INTEL_SPEED_SELECT_CONFIG_2
```

To trigger the Intel SST-PP configuration during node provisioning, one of the traits can be added to the flavor.

```
openstack flavor set baremetal --property trait:CUSTOM_INTEL_SPEED_SELECT_
↪CONFIG_1=required
```

Finally create a server with `baremetal` flavor to provision a baremetal node with Intel SST-PP profile *Config 1*.

IPMI driver

Overview

The `ipmi` hardware type manage nodes by using **IPMI** (Intelligent Platform Management Interface) protocol versions 2.0 or 1.5. It uses the `IPMITool` utility which is an open-source command-line interface (CLI) for controlling IPMI-enabled devices.

Glossary

- **IPMI** - Intelligent Platform Management Interface.
- **IPMB** - Intelligent Platform Management Bus/Bridge.
- **BMC** - Baseboard Management Controller.
- **RMCP** - Remote Management Control Protocol.

Enabling the IPMI hardware type

Please see *Configuring IPMI support* for the required dependencies.

1. The `ipmi` hardware type is enabled by default starting with the Ocata release. To enable it explicitly, add the following to your `ironic.conf`:

```
[DEFAULT]
enabled_hardware_types = ipmi
enabled_management_interfaces = ipmitool,noop
enabled_power_interfaces = ipmitool
```

Optionally, enable the *vendor passthru interface* and either or both *console interfaces*:

```
[DEFAULT]
enabled_hardware_types = ipmi
enabled_console_interfaces = ipmitool-socat,ipmitool-shellinabox,no-
↪console
enabled_management_interfaces = ipmitool,noop
enabled_power_interfaces = ipmitool
enabled_vendor_interfaces = ipmitool,no-vendor
```

2. Restart the Ironic conductor service.

Please see *Enabling drivers and hardware types* for more details.

Registering a node with the IPMI driver

Nodes configured to use the IPMItool drivers should have the `driver` field set to `ipmi`.

The following configuration value is required and has to be added to the nodes `driver_info` field:

- `ipmi_address`: The IP address or hostname of the BMC.

Other options may be needed to match the configuration of the BMC, the following options are optional, but in most cases, its considered a good practice to have them set:

- `ipmi_username`: The username to access the BMC; defaults to `NULL` user.
- `ipmi_password`: The password to access the BMC; defaults to `NULL`.
- `ipmi_port`: The remote IPMI RMCP port. By default ipmitool will use the port `623`.

Note: It is highly recommend that you setup a username and password for your BMC.

The `baremetal node create` command can be used to enroll a node with an IPMItool-based driver. For example:

```
baremetal node create --driver ipmi \  
  --driver-info ipmi_address=<address> \  
  --driver-info ipmi_username=<username> \  
  --driver-info ipmi_password=<password>
```

Advanced configuration

When a simple configuration such as providing the address, username and password is not enough, the IPMItool driver contains many other options that can be used to address special usages.

Single/Double bridging functionality

Note: A version of IPMItool higher or equal to 1.8.12 is required to use the bridging functionality.

There are two different bridging functionalities supported by the IPMItool-based drivers: *single* bridge and *dual* bridge.

The following configuration values need to be added to the nodes `driver_info` field so bridging can be used:

- `ipmi_bridging`: The bridging type; default is *no*; other supported values are *single* for single bridge or *dual* for double bridge.
- **`ipmi_local_address`: The local IPMB address for bridged requests.** Required only if `ipmi_bridging` is set to *single* or *dual*. This configuration is optional, if not specified it will be auto discovered by IPMItool.
- `ipmi_target_address`: The destination address for bridged requests. Required only if `ipmi_bridging` is set to *single* or *dual*.
- `ipmi_target_channel`: The destination channel for bridged requests. Required only if `ipmi_bridging` is set to *single* or *dual*.

Double bridge specific options:

- `ipmi_transit_address`: The transit address for bridged requests. Required only if `ipmi_bridging` is set to *dual*.
- `ipmi_transit_channel`: The transit channel for bridged requests. Required only if `ipmi_bridging` is set to *dual*.

The parameter `ipmi_bridging` should specify the type of bridging required: *single* or *dual* to access the bare metal node. If the parameter is not specified, the default value will be set to *no*.

The `baremetal node set` command can be used to set the required bridging information to the Ironic node enrolled with the IPMItool driver. For example:

- Single Bridging:

```
baremetal node set <UUID or name> \
  --driver-info ipmi_local_address=<address> \
  --driver-info ipmi_bridging=single \
  --driver-info ipmi_target_channel=<channel> \
  --driver-info ipmi_target_address=<target address>
```

- Double Bridging:

```
baremetal node set <UUID or name> \
  --driver-info ipmi_local_address=<address> \
  --driver-info ipmi_bridging=dual \
  --driver-info ipmi_transit_channel=<transit channel> \
  --driver-info ipmi_transit_address=<transit address> \
  --driver-info ipmi_target_channel=<target channel> \
  --driver-info ipmi_target_address=<target address>
```

Changing the version of the IPMI protocol

The IPMITool-based drivers works with the versions 2.0 and 1.5 of the IPMI protocol. By default, the version 2.0 is used.

In order to change the IPMI protocol version in the bare metal node, the following option needs to be set to the nodes `driver_info` field:

- `ipmi_protocol_version`: The version of the IPMI protocol; default is 2.0. Supported values are 1.5 or 2.0.

The `baremetal node set` command can be used to set the desired protocol version:

```
baremetal node set <UUID or name> --driver-info ipmi_protocol_version=
↔<version>
```

Warning: Version 1.5 of the IPMI protocol does not support encryption. Therefore, it is highly recommended that version 2.0 is used.

Cipher suites

IPMI 2.0 introduces support for encryption and allows setting which cipher suite to use. Traditionally, `ipmitool` was using cipher suite 3 by default, but since SHA1 no longer complies with modern security requirement, recent versions (e.g. the one used in RHEL 8.2) are switching to suite 17.

Normally, the cipher suite to use is negotiated with the BMC using the special command. On some hardware the negotiation yields incorrect results and IPMI commands fail with

```
Error in open session response message : no matching cipher suite
Error: Unable to establish IPMI v2 / RMCP+ session
```

Another possible problem is `ipmitool` commands taking very long (tens of seconds or even minutes) because the BMC does not support cipher suite negotiation. In both cases you can specify the required suite yourself, e.g.:

```
baremetal node set <UUID or name> --driver-info ipmi_cipher_suite=3
```

In scenarios where the operator can't specify the `ipmi_cipher_suite` for each node, the configuration parameter `[ipmi]/cipher_suite_versions` can be set to a list of cipher suites that will be used, Ironic will attempt to find a value that can be used from the list provided (from last to first):

```
[ipmi]
cipher_suite_versions = ['1','2','3','6','7','8','11','12']
```

To find the suitable values for this configuration, you can check the field *RMCP+ Cipher Suites* after running an `ipmitool` command, e.g:

```
$ ipmitool -I lanplus -H $HOST -U $USER -v -R 12 -N 5 lan print
# output
Set in Progress      : Set Complete
Auth Type Support    : NONE MD2 MD5 PASSWORD OEM
Auth Type Enable     : Callback : NONE MD2 MD5 PASSWORD OEM
IP Address Source    : Static Address
IP Address           : <IP>
Subnet Mask          : <Subnet>
MAC Address          : <MAC>
RMCP+ Cipher Suites : 0,1,2,3,6,7,8,11,12
```

Static boot order configuration

See *Static boot order configuration*.

Vendor Differences

While the Intelligent Platform Management Interface (IPMI) interface is based upon a defined standard, the Ironic community is aware of at least one vendor which utilizes a non-standard boot device selector. In essence, this could be something as simple as different interpretation of the standard.

As of October 2020, the known difference is with Supermicro hardware where a selector of `0x24`, signifying a *REMOTE* boot device in the standard, must be used when a boot operation from the local disk subsystem is requested **in UEFI mode**. This is contrary to BIOS mode where the same BMCs expect the selector to be a value of `0x08`.

Because the BMC does not respond with any sort of error, nor do we want to risk BMC connectivity issues by explicitly querying all BMCs what vendor it may be before every operation, the vendor can automatically be recorded in the `properties` field `vendor`. When this is set to a value of `supermicro`, Ironic will navigate the UEFI behavior difference enabling the UEFI to be requested with boot to disk.

Example:

```
baremetal node set <UUID or name> \
  --properties vendor="supermicro"
```

Luckily, Ironic will attempt to perform this detection in power synchronization process, and record this value if not already set.

While similar issues may exist when setting the boot mode and target boot device in other vendors BMCs, we are not aware of them at present. Should you encounter such an issue, please feel free to report this via [Storyboard](#), and be sure to include the `chassis bootparam get 5` output value along with the `mc info` output from your BMC.

Example:

```
ipmitool -I lanplus -H <BMC ADDRESS> -U <Username> -P <Password> \
mc info
ipmitool -I lanplus -H <BMC ADDRESS> -U <Username> -P <Password> \
chassis bootparam get 5
```

iRMC driver

Overview

The iRMC driver enables control FUJITSU PRIMERGY via ServerView Common Command Interface (SCCI). Support for FUJITSU PRIMERGY servers consists of the `irmc` hardware type and a few hardware interfaces specific for that hardware type.

Prerequisites

- Install `python-scciclient` and `pysnmp` packages:

```
$ pip install "python-scciclient>=0.7.2" pysnmp
```

Hardware Type

The `irmc` hardware type is available for FUJITSU PRIMERGY servers. For information on how to enable the `irmc` hardware type, see [Enabling hardware types](#).

Hardware interfaces

The `irmc` hardware type overrides the selection of the following hardware interfaces:

- **bios** Supports `irmc` and `no-bios`. The default is `irmc`.
- **boot** Supports `irmc-virtual-media`, `irmc-pxe`, and `pxe`. The default is `irmc-virtual-media`. The `irmc-virtual-media` boot interface enables the virtual media based deploy with IPA (Ironic Python Agent).

Warning: We deprecated the `pxe` boot interface when used with `irmc` hardware type. Support for this interface will be removed in the future. Instead, use `irmc-pxe`. `irmc-pxe` boot interface was introduced in Pike.

- **console** Supports `ipmitool-socat`, `ipmitool-shellinbox`, and `no-console`. The default is `ipmitool-socat`.

- **inspect** Supports `irmc`, `inspector`, and `no-inspect`. The default is `irmc`.

Note: `Ironic Inspector` needs to be present and configured to use `inspector` as the `inspect` interface.

- **management** Supports only `irmc`.
- **power** Supports `irmc`, which enables power control via ServerView Common Command Interface (SCCI), by default. Also supports `ipmitool`.
- **raid** Supports `irmc`, `no-raid` and `agent`. The default is `no-raid`.

For other hardware interfaces, `irmc` hardware type supports the Bare Metal reference interfaces. For more details about the hardware interfaces and how to enable the desired ones, see [Enabling hardware interfaces](#).

Here is a complete configuration example with most of the supported hardware interfaces enabled for `irmc` hardware type.

```
[DEFAULT]
enabled_hardware_types = irmc
enabled_bios_interfaces = irmc
enabled_boot_interfaces = irmc-virtual-media,irmc-pxe
enabled_console_interfaces = ipmitool-socat,ipmitool-shellinabox,no-console
enabled_deploy_interfaces = direct
enabled_inspect_interfaces = irmc,inspector,no-inspect
enabled_management_interfaces = irmc
enabled_network_interfaces = flat,neutron
enabled_power_interfaces = irmc
enabled_raid_interfaces = no-raid,irmc
enabled_storage_interfaces = noop,cinder
enabled_vendor_interfaces = no-vendor,ipmitool
```

Here is a command example to enroll a node with `irmc` hardware type.

```
baremetal node create \
  --bios-interface irmc \
  --boot-interface irmc-pxe \
  --deploy-interface direct \
  --inspect-interface irmc \
  --raid-interface irmc
```

Node configuration

- Each node is configured for `irmc` hardware type by setting the following ironic node objects properties:
 - `driver_info/irmc_address` property to be IP address or hostname of the iRMC.
 - `driver_info/irmc_username` property to be username for the iRMC with administrator privileges.
 - `driver_info/irmc_password` property to be password for `irmc_username`.

- `properties/capabilities` property to be `boot_mode:uefi` if UEFI boot is required.
- `properties/capabilities` property to be `secure_boot:true` if UEFI Secure Boot is required. Please refer to *UEFI Secure Boot Support* for more information.
- The following properties are also required if `irmc-virtual-media` boot interface is used:
 - `driver_info/irmc_deploy_iso` property to be either deploy iso file name, Glance UUID, or Image Service URL.
 - `instance info/irmc_boot_iso` property to be either boot iso file name, Glance UUID, or Image Service URL. This is optional property when `boot_option` is set to `netboot`.
- All of the nodes are configured by setting the following configuration options in the `[irmc]` section of `/etc/ironic/ironic.conf`:
 - `port`: Port to be used for iRMC operations; either 80 or 443. The default value is 443. Optional.
 - `auth_method`: Authentication method for iRMC operations; either `basic` or `digest`. The default value is `basic`. Optional.
 - `client_timeout`: Timeout (in seconds) for iRMC operations. The default value is 60. Optional.
 - `sensor_method`: Sensor data retrieval method; either `ipmi tool` or `scsi`. The default value is `ipmitool`. Optional.
- The following options are required if `irmc-virtual-media` boot interface is enabled:
 - `remote_image_share_root`: Ironic conductor nodes NFS or CIFS root path. The default value is `/remote_image_share_root`.
 - `remote_image_server`: IP of remote image server.
 - `remote_image_share_type`: Share type of virtual media, either NFS or CIFS. The default is CIFS.
 - `remote_image_share_name`: share name of `remote_image_server`. The default value is `share`.
 - `remote_image_user_name`: User name of `remote_image_server`.
 - `remote_image_user_password`: Password of `remote_image_user_name`.
 - `remote_image_user_domain`: Domain name of `remote_image_user_name`.
- The following options are required if `irmc inspect` interface is enabled:
 - `snmp_version`: SNMP protocol version; either `v1`, `v2c` or `v3`. The default value is `v2c`. Optional.
 - `snmp_port`: SNMP port. The default value is 161. Optional.
 - `snmp_community`: SNMP community required for versions `v1` and `v2c`. The default value is `public`. Optional.
 - `snmp_security`: SNMP security name required for version `v3`. Optional.
- Each node can be further configured by setting the following ironic node objects properties which override the parameter values in `[irmc]` section of `/etc/ironic/ironic.conf`:
 - `driver_info/irmc_port` property overrides `port`.

- driver_info/irmc_auth_method property overrides auth_method.
- driver_info/irmc_client_timeout property overrides client_timeout.
- driver_info/irmc_sensor_method property overrides sensor_method.
- driver_info/irmc_snmp_version property overrides snmp_version.
- driver_info/irmc_snmp_port property overrides snmp_port.
- driver_info/irmc_snmp_community property overrides snmp_community.
- driver_info/irmc_snmp_security property overrides snmp_security.

Optional functionalities for the `irmc` hardware type

UEFI Secure Boot Support

The hardware type `irmc` supports secure boot deploy, see *UEFI secure boot mode* for details.

Warning: Secure boot feature is not supported with `pxe` boot interface.

Node Cleaning Support

The `irmc` hardware type supports node cleaning. For more information on node cleaning, see *Node cleaning*.

Supported Automated Cleaning Operations

The automated cleaning operations supported are:

- `restore_irmc_bios_config`: Restores BIOS settings on a baremetal node from backup data. If this clean step is enabled, the BIOS settings of a baremetal node will be backed up automatically before the deployment. By default, this clean step is disabled with priority `0`. Set its priority to a positive integer to enable it. The recommended value is `10`.

Warning: `pxe` boot interface, when used with `irmc` hardware type, does not support this clean step. If uses `irmc` hardware type, it is required to select `irmc-pxe` or `irmc-virtual-media` as the boot interface in order to make this clean step work.

Configuration options for the automated cleaning steps are listed under `[irmc]` section in `ironic.conf`

```
clean_priority_restore_irmc_bios_config = 0
```

For more information on node automated cleaning, see *Automated cleaning*

Boot from Remote Volume

The `irmc` hardware type supports the generic iPXE-based remote volume booting when using the following boot interfaces:

- `irmc-pxe`
- `pxe`

In addition, the `irmc` hardware type supports remote volume booting without iPXE. This is available when using the `irmc-virtual-media` boot interface. This feature configures a node to boot from a remote volume by using the API of iRMC. It supports iSCSI and FibreChannel.

Configuration

In addition to the configuration for generic drivers to *remote volume boot*, the iRMC driver requires the following configuration:

- It is necessary to set physical port IDs to network ports and volume connectors. All cards including those not used for volume boot should be registered.

The format of a physical port ID is: `<Card Type><Slot No>-<Port No>` where:

- `<Card Type>`: could be LAN, FC or CNA
- `<Slot No>`: 0 indicates onboard slot. Use 1 to 9 for add-on slots.
- `<Port No>`: A port number starting from 1.

These IDs are specified in a nodes `driver_info[irmc_pci_physical_ids]`. This value is a dictionary. The key is the UUID of a resource (Port or Volume Connector) and its value is the physical port ID. For example:

```
{
  "1ecd14ee-c191-4007-8413-16bb5d5a73a2": "LAN0-1",
  "87f6c778-e60e-4df2-bdad-2605d53e6fc0": "CNA1-1"
}
```

It can be set with the following command:

```
baremetal node set $NODE_UUID \
--driver-info irmc_pci_physical_ids={} \
--driver-info irmc_pci_physical_ids/$PORT_UUID=LAN0-1 \
--driver-info irmc_pci_physical_ids/$VOLUME_CONNECTOR_UUID=CNA1-1
```

- For iSCSI boot, volume connectors with both types `iqn` and `ip` are required. The configuration with DHCP is not supported yet.
- For iSCSI, the size of the storage network is needed. This value should be specified in a nodes `driver_info[irmc_storage_network_size]`. It must be a positive integer < 32. For example, if the storage network is 10.2.0.0/22, use the following command:

```
baremetal node set $NODE_UUID --driver-info irmc_storage_network_size=22
```

Supported hardware

The driver supports the PCI controllers, Fibrechannel Cards, Converged Network Adapters supported by Fujitsu ServerView Virtual-IO Manager.

Hardware Inspection Support

The `irmc` hardware type provides the iRMC-specific hardware inspection with `irmc inspect` interface.

Note: SNMP requires being enabled in ServerView's iRMC S4 Web Server(Network SettingsSNMP section).

Configuration

The Hardware Inspection Support in the iRMC driver requires the following configuration:

- It is necessary to set ironic configuration with `gpu_ids` and `fpga_ids` options in `[irmc]` section. `gpu_ids` and `fpga_ids` are lists of `<vendorID>/<deviceID>` where:
 - `<vendorID>`: 4 hexadecimal digits starts with 0x.
 - `<deviceID>`: 4 hexadecimal digits starts with 0x.

Here are sample values for `gpu_ids` and `fpga_ids`:

```
gpu_ids = 0x1000/0x0079,0x2100/0x0080
fpga_ids = 0x1000/0x005b,0x1100/0x0180
```

- The `python-ssciclient` package requires `pyghmi` version `>= 1.0.22` and `pysnmp` version `>= 4.2.3`. They are used by the conductor service on the conductor. The latest version of `pyghmi` can be downloaded from [here](#) and `pysnmp` can be downloaded from [here](#).

Supported properties

The inspection process will discover the following essential properties (properties required for scheduling deployment):

- `memory_mb`: memory size
- `cpus`: number of cpus
- `cpu_arch`: cpu architecture
- `local_gb`: disk size

Inspection can also discover the following extra capabilities for iRMC driver:

- `irmc_firmware_version`: iRMC firmware version
- `rom_firmware_version`: ROM firmware version

- `trusted_boot`: The flag whether TPM(Trusted Platform Module) is supported by the server. The possible values are True or False.
- `server_model`: server model
- `pci_gpu_devices`: number of gpu devices connected to the bare metal.

Inspection can also set/unset nodes traits with the following cpu type for iRMC driver:

- `CUSTOM_CPU_FPGA`: The bare metal contains fpga cpu type.

Note:

- The disk size is returned only when eLCM License for FUJITSU PRIMERGY servers is activated. If the license is not activated, then Hardware Inspection will fail to get this value.
 - Before inspecting, if the server is power-off, it will be turned on automatically. System will wait for a few second before start inspecting. After inspection, power status will be restored to the previous state.
-

The operator can specify these capabilities in compute service flavor, for example:

```
openstack flavor set baremetal-flavor-name --property capabilities:irmc_
↪firmware_version="iRMC S4-8.64F"

openstack flavor set baremetal-flavor-name --property capabilities:server_
↪model="TX2540M1F5"

openstack flavor set baremetal-flavor-name --property capabilities:pci_gpu_
↪devices="1"
```

See *Capabilities discovery* for more details and examples.

The operator can add a trait in compute service flavor, for example:

```
baremetal node add trait $NODE_UUID CUSTOM_CPU_FPGA
```

A valid trait must be no longer than 255 characters. Standard traits are defined in the `os_traits` library. A custom trait must start with the prefix `CUSTOM_` and use the following characters: A-Z, 0-9 and `_`.

RAID configuration Support

The `irmc` hardware type provides the iRMC RAID configuration with `irmc raid` interface.

Note:

- RAID implementation for `irmc` hardware type is based on eLCM license and SDCard. Otherwise, SP(Service Platform) in lifecycle management must be available.
 - RAID implementation only supported for RAIDAdapter 0 in Fujitsu Servers.
-

Configuration

The RAID configuration Support in the iRMC drivers requires the following configuration:

- It is necessary to set ironic configuration into Node with JSON file option:

```
$ baremetal node set <node-uuid-or-name> \  
--target-raid-config <JSON file containing target RAID configuration>
```

Here is some sample values for JSON file:

```
{  
  "logical_disks": [  
    {  
      "size_gb": 1000,  
      "raid_level": "1"  
    }  
  ]  
}
```

or:

```
{  
  "logical_disks": [  
    {  
      "size_gb": 1000,  
      "raid_level": "1",  
      "controller": "FTS RAID Ctrl SAS 6G 1GB (D3116C) (0)",  
      "physical_disks": [  
        "0",  
        "1"  
      ]  
    }  
  ]  
}
```

Note: RAID 1+0 and 5+0 in iRMC driver does not support property `physical_disks` in `target_raid_config` during create raid configuration yet. See following example:

```
{  
  "logical_disks":  
  [  
    {  
      "size_gb": "MAX",  
      "raid_level": "1+0"  
    }  
  ]  
}
```

See *RAID Configuration* for more details and examples.

Supported properties

The RAID configuration using iRMC driver supports following parameters in JSON file:

- `size_gb`: is mandatory properties in Ironic.
- `raid_level`: is mandatory properties in Ironic. Currently, iRMC Server supports following RAID levels: 0, 1, 5, 6, 1+0 and 5+0.
- `controller`: is name of the controller as read by the RAID interface.
- `physical_disks`: are specific values for each raid array in LogicalDrive which operator want to set them along with `raid_level`.

The RAID configuration is supported as a manual cleaning step.

Note:

- iRMC server will power-on after create/delete raid configuration is applied, FGI (Foreground Initialize) will process raid configuration in iRMC server, thus the operation will completed upon power-on and power-off when created RAID on iRMC server.

See *RAID Configuration* for more details and examples.

BIOS configuration Support

The `irmc` hardware type provides the iRMC BIOS configuration with `irmc bios` interface.

Warning: `irmc bios` interface does not support `factory_reset`.

Configuration

The BIOS configuration in the iRMC driver supports the following settings:

- `boot_option_filter`: Specifies from which drives can be booted. This supports following options: `UefiAndLegacy`, `LegacyOnly`, `UefiOnly`.
- `check_controllers_health_status_enabled`: The UEFI FW checks the controller health status. This supports following options: `true`, `false`.
- `cpu_active_processor_cores`: The number of active processor cores 1n. Option 0 indicates that all available processor cores are active.
- `cpu_adjacent_cache_line_prefetch_enabled`: The processor loads the requested cache line and the adjacent cache line. This supports following options: `true`, `false`.
- `cpu_vt_enabled`: Supports the virtualization of platform hardware and several software environments, based on Virtual Machine Extensions to support the use of several software environments using virtual computers. This supports following options: `true`, `false`.
- `flash_write_enabled`: The system BIOS can be written. Flash BIOS update is possible. This supports following options: `true`, `false`.

- `hyper_threading_enabled`: Hyper-threading technology allows a single physical processor core to appear as several logical processors. This supports following options: `true`, `false`.
- `keep_void_boot_options_enabled`: Boot Options will not be removed from Boot Option Priority list. This supports following options: `true`, `false`.
- `launch_csm_enabled`: Specifies whether the Compatibility Support Module (CSM) is executed. This supports following options: `true`, `false`.
- `os_energy_performance_override_enabled`: Prevents the OS from overruling any energy efficiency policy setting of the setup. This supports following options: `true`, `false`.
- `pci_aspm_support`: Active State Power Management (ASPM) is used to power-manage the PCI Express links, thus consuming less power. This supports following options: `Disabled`, `Auto`, `L0Limited`, `L1only`, `L0Force`.
- `pci_above_4g_decoding_enabled`: Specifies if memory resources above the 4GB address boundary can be assigned to PCI devices. This supports following options: `true`, `false`.
- `power_on_source`: Specifies whether the switch on sources for the system are managed by the BIOS or the ACPI operating system. This supports following options: `BiosControlled`, `AcpiControlled`.
- `single_root_io_virtualization_support_enabled`: Single Root IO Virtualization Support is active. This supports following options: `true`, `false`.

The BIOS configuration is supported as a manual cleaning step. See *BIOS Configuration* for more details and examples.

Supported platforms

This driver supports FUJITSU PRIMERGY BX S4 or RX S8 servers and above.

- PRIMERGY BX920 S4
- PRIMERGY BX924 S4
- PRIMERGY RX300 S8

When `irmc` power interface is used, Soft Reboot (Graceful Reset) and Soft Power Off (Graceful Power Off) are only available if `ServerView` agents are installed. See *iRMC S4 Manual* for more details.

RAID configuration feature supports FUJITSU PRIMERGY servers with RAID-Ctrl-SAS-6G-1GB(D3116C) controller and above. For detail supported controller with OOB-RAID configuration, please see the *whitepaper for iRMC RAID configuration*.

Redfish driver

Overview

The `redfish` driver enables managing servers compliant with the *Redfish* protocol.

Prerequisites

- The `Sushy` library should be installed on the ironic conductor node(s).

For example, it can be installed with `pip`:

```
sudo pip install sushy
```

Enabling the Redfish driver

1. Add `redfish` to the list of `enabled_hardware_types`, `enabled_power_interfaces`, `enabled_management_interfaces` and `enabled_inspect_interfaces` as well as `redfish-virtual-media` to `enabled_boot_interfaces` in `/etc/ironic/ironic.conf`. For example:

```
[DEFAULT]
...
enabled_hardware_types = ipmi,redfish
enabled_boot_interfaces = ipxe,redfish-virtual-media
enabled_power_interfaces = ipmitool,redfish
enabled_management_interfaces = ipmitool,redfish
enabled_inspect_interfaces = inspector,redfish
```

2. Restart the ironic conductor service:

```
sudo service ironic-conductor restart

# Or, for RDO:
sudo systemctl restart openstack-ironic-conductor
```

Registering a node with the Redfish driver

Nodes configured to use the driver should have the `driver` property set to `redfish`.

The following properties are specified in the nodes `driver_info` field:

- **redfish_address:** The URL address to the Redfish controller. It must include the authority portion of the URL, and can optionally include the scheme. If the scheme is missing, `https` is assumed. For example: `https://mgmt.vendor.com`. This is required.
- **redfish_system_id:** The canonical path to the `ComputerSystem` resource that the driver will interact with. It should include the root service, version and the unique resource path to the `ComputerSystem`. This property is only required if target BMC manages more than one `ComputerSystem`. Otherwise ironic will pick the only available `ComputerSystem` automatically. For example: `/redfish/v1/Systems/1`.
- **redfish_username:** User account with admin/server-profile access privilege. Although not required, it is highly recommended.
- **redfish_password:** User account password. Although not required, it is highly recommended.

- **redfish_verify_ca:** If `redfish_address` has the `https` scheme, the driver will use a secure (TLS) connection when talking to the Redfish controller. By default (if this is not set or set to `True`), the driver will try to verify the host certificates. This can be set to the path of a certificate file or directory with trusted certificates that the driver will use for verification. To disable verifying TLS, set this to `False`. This is optional.
- **redfish_auth_type:** Redfish HTTP client authentication method. Can be `basic`, `session` or `auto`. The `auto` mode first tries `session` and falls back to `basic` if `session` authentication is not supported by the Redfish BMC. Default is set in `ironic config` as `[redfish]auth_type`.

The `baremetal node create` command can be used to enroll a node with the `redfish` driver. For example:

```
baremetal node create --driver redfish --driver-info \  
redfish_address=https://example.com --driver-info \  
redfish_system_id=/redfish/v1/Systems/CX34R87 --driver-info \  
redfish_username=admin --driver-info redfish_password=password \  
--name node-0
```

For more information about enrolling nodes see *Enrollment* in the install guide.

Boot mode support

The `redfish` hardware type can read current boot mode from the bare metal node as well as set it to either Legacy BIOS or UEFI.

Note: Boot mode management is the optional part of the Redfish specification. Not all Redfish-compliant BMCs might implement it. In that case it remains the responsibility of the operator to configure proper boot mode to their bare metal nodes.

UEFI secure boot

Secure boot mode can be automatically set and unset during deployment for nodes in UEFI boot mode, see *UEFI secure boot mode* for an explanation how to use it.

Two clean and deploy steps are provided for key management:

management.reset_secure_boot_keys_to_default resets secure boot keys to their manufacturing defaults.

management.clear_secure_boot_keys removes all secure boot keys from the node.

Out-Of-Band inspection

The `redfish` hardware type can inspect the bare metal node by querying Redfish compatible BMC. This process is quick and reliable compared to the way the `inspector` hardware type works i.e. booting bare metal node into the introspection ramdisk.

Note: The `redfish` inspect interface relies on the optional parts of the Redfish specification. Not all Redfish-compliant BMCs might serve the required information, in which case bare metal node inspection will fail.

Note: The `local_gb` property cannot always be discovered, for example, when a node does not have local storage or the Redfish implementation does not support the required schema. In this case the property will be set to 0.

Virtual media boot

The idea behind virtual media boot is that BMC gets hold of the boot image one way or the other (e.g. by HTTP GET, other methods are defined in the standard), then inserts it into nodes virtual drive as if it was burnt on a physical CD/DVD. The node can then boot from that virtual drive into the operating system residing on the image.

The major advantage of virtual media boot feature is that potentially unreliable TFTP image transfer phase of PXE protocol suite is fully eliminated.

Hardware types based on the `redfish` fully support booting deploy/rescue and user images over virtual media. Ironic builds bootable ISO images, for either UEFI or BIOS (Legacy) boot modes, at the moment of node deployment out of kernel and ramdisk images associated with the ironic node.

To boot a node managed by `redfish` hardware type over virtual media using BIOS boot mode, it suffice to set ironic boot interface to `redfish-virtual-media`, as opposed to `ipmitool`.

```
baremetal node set --boot-interface redfish-virtual-media node-0
```

Warning: Dell hardware requires a non-standard Redfish call to boot from virtual media, thus you **must** use the `idrac` hardware type and the `idrac-redfish-virtual-media` boot interface with it instead. See *iDRAC driver* for more details on this hardware type.

If UEFI boot mode is desired, the user should additionally supply EFI System Partition image (**ESP**), see *Configuring an ESP image* for details.

If `[driver_info]/config_via_floppy` boolean property of the node is set to `true`, ironic will create a file with runtime configuration parameters, place into on a FAT image, then insert the image into nodes virtual floppy drive.

When booting over PXE or virtual media, and user instance requires some specific kernel configuration, the nodes `instance_info[kernel_append_params]` or `driver_info[kernel_append_params]` properties can be used to pass user-specified kernel command line parameters.

```
baremetal node set node-0 \  
  --driver-info kernel_append_params="nofb nomodeset vga=normal"
```

Note: The `driver_info` field is supported starting with the Xena release.

For ramdisk boot, the `instance_info[ramdisk_kernel_arguments]` property serves the same purpose.

Pre-built ISO images

By default an ISO image is built per node using the deploy kernel and initramfs provided in the configuration or the nodes `driver_info`. Starting with the Wallaby release its possible to provide a pre-built ISO image:

```
baremetal node set node-0 \  
  --driver_info deploy_iso=http://url/of/deploy.iso \  
  --driver_info rescue_iso=http://url/of/rescue.iso
```

Note: OpenStack Image service (glance) image IDs and `file://` links are also accepted.

Note: Before the Xena release the parameters were called `redfish_deploy_iso` and `redfish_rescue_iso` accordingly. The old names are still supported for backward compatibility.

No customization is currently done to the image, so e.g. *Layer 3 or DHCP-less ramdisk booting* wont work. *Configuring an ESP image* is also unnecessary.

Configuring an ESP image

An ESP image is an image that contains the necessary bootloader to boot the ISO in UEFI mode. You will need a GRUB2 image file, as well as Shim for secure boot. See *UEFI PXE - Grub setup* for an explanation how to get them.

Then the following script can be used to build an ESP image:

```
DEST=/path/to/esp.img  
GRUB2=/path/to/grub.efi  
SHIM=/path/to/shim.efi  
TEMP_MOUNT=$(mktemp -d)  
  
dd if=/dev/zero of=$DEST bs=4096 count=1024  
mkfs.fat -s 4 -r 512 -S 4096 $DEST  
  
sudo mount $DEST $TEMP_MOUNT  
sudo mkdir -p $DEST/EFI/BOOT  
sudo cp "$SHIM" $DEST/EFI/BOOT/BOOTX64.efi
```

(continues on next page)

(continued from previous page)

```
sudo cp "$GRUB2" $DEST/EFI/BOOT/GRUBX64.efi
sudo umount $TEMP_MOUNT
```

Note: If you use an architecture other than x86-64, you'll need to adjust the destination paths.

The resulting image should be provided via the `driver_info/bootloader` ironic node property in form of an image UUID or a URL:

```
baremetal node set --driver-info bootloader=<glance-uuid-or-url> node-0
```

Alternatively, set the bootloader UUID or URL in the configuration file:

```
[conductor]
bootloader = <glance-uuid-or-url>
```

Finally, you need to provide the correct GRUB2 configuration path for your image. In most cases this path will depend on your distribution, more precisely, the distribution you took the GRUB2 image from. For example:

CentOS:

```
[DEFAULT]
grub_config_path = EFI/centos/grub.cfg
```

Ubuntu:

```
[DEFAULT]
grub_config_path = EFI/ubuntu/grub.cfg
```

Note: Unlike in the script above, these paths are case-sensitive!

Virtual Media Ramdisk

The `ramdisk` deploy interface can be used in concert with the `redfish-virtual-media` boot interface to facilitate the boot of a remote node utilizing pre-supplied virtual media. See *Booting a Ramdisk or an ISO* for information on how to enable and configure it.

Instead of supplying an `[instance_info]/image_source` parameter, a `[instance_info]/boot_iso` parameter can be supplied. The image will be downloaded by the conductor, and the instance will be booted using the supplied ISO image. In accordance with the `ramdisk` deployment interface behavior, once booted the machine will have a `provision_state` of `ACTIVE`.

```
baremetal node set <node name or UUID> \
  --boot-interface redfish-virtual-media \
  --deploy-interface ramdisk \
  --instance_info boot_iso=http://url/to.iso
```

This initial interface does not support bootloader configuration parameter injection, as such the `[instance_info]/kernel_append_params` setting is ignored.

Configuration drives are supported starting with the Wallaby release for nodes that have a free virtual USB slot:

```
baremetal node deploy <node name or UUID> \  
  --config-drive '{"meta_data": {...}, "user_data": "..."}'
```

or via a link to a raw image:

```
baremetal node deploy <node name or UUID> \  
  --config-drive http://example.com/config.img
```

Layer 3 or DHCP-less ramdisk booting

DHCP-less deploy is supported by the Redfish virtual media boot. See *Layer 3 or DHCP-less ramdisk booting* for more information.

Firmware update using manual cleaning

The `redfish` hardware type supports updating the firmware on nodes using a manual cleaning step.

The firmware update cleaning step allows one or more firmware updates to be applied to a node. If multiple updates are specified, then they are applied sequentially in the order given. The server is rebooted once per update. If a failure occurs, the cleaning step immediately fails which may result in some updates not being applied. If the node is placed into maintenance mode while a firmware update cleaning step is running that is performing multiple firmware updates, the update in progress will complete, and processing of the remaining updates will pause. When the node is taken out of maintenance mode, processing of the remaining updates will continue.

When updating the BMC firmware, the BMC may become unavailable for a period of time as it resets. In this case, it may be desirable to have the cleaning step wait after the update has been applied before indicating that the update was successful. This allows the BMC time to fully reset before further operations are carried out against it. To cause the cleaning step to wait after applying an update, an optional `wait` argument may be specified in the firmware image dictionary. The value of this argument indicates the number of seconds to wait following the update. If the `wait` argument is not specified, then this is equivalent to `wait 0`, meaning that it will not wait and immediately proceed with the next firmware update if there is one, or complete the cleaning step if not.

The `update_firmware` cleaning step accepts JSON in the following format:

```
[{  
  "interface": "management",  
  "step": "update_firmware",  
  "args": {  
    "firmware_images": [  
      {  
        "url": "<url_to_firmware_image1>",  
        "wait": <number_of_seconds_to_wait>  
      },  
      {  
        "url": "<url_to_firmware_image2>",  
        "wait": <number_of_seconds_to_wait>  
      }  
    ]  
  }  
}]
```

(continues on next page)

(continued from previous page)

```

    {
        "url": "<url_to_firmware_image2>"
    },
    ...
]
}
}]

```

The different attributes of the `update_firmware` cleaning step are as follows:

Attribute	Description
<code>interface</code>	Interface of the cleaning step. Must be management for firmware update
<code>step</code>	Name of cleaning step. Must be <code>update_firmware</code> for firmware update
<code>args</code>	Keyword-argument entry (<code><name>: <value></code>) being passed to cleaning step
<code>args.firmware_images</code>	Ordered list of dictionaries of firmware images to be applied

Each firmware image dictionary, is of the form:

```

{
  "url": "<URL of firmware image file>",
  "wait": <Optional time in seconds to wait after applying update>
}

```

The `url` argument in the firmware image dictionary is mandatory, while the `wait` argument is optional.

Note: Only `http` and `https` URLs are currently supported in the `url` argument.

Note: At the present time, targets for the firmware update cannot be specified. In testing, the BMC applied the update to all applicable targets on the node. It is assumed that the BMC knows what components a given firmware image is applicable to.

To perform a firmware update, first download the firmware to a web server that the BMC has network access to. This could be the ironic conductor web server or another web server on the BMC network. Using a web browser, `curl`, or similar tool on a server that has network access to the BMC, try downloading the firmware to verify that the URLs are correct and that the web server is configured properly.

Next, construct the JSON for the firmware update cleaning step to be executed. When launching the firmware update, the JSON may be specified on the command line directly or in a file. The following example shows one cleaning step that installs two firmware updates. The first updates the BMC firmware followed by a five minute wait to allow the BMC time to start back up. The second updates the firmware on all applicable NICs.:

```

[ {
  "interface": "management",
  "step": "update_firmware",
  "args": {

```

(continues on next page)

(continued from previous page)

```
"firmware_images":[
  {
    "url": "http://192.0.2.10/BMC_4_22_00_00.EXE",
    "wait": 300
  },
  {
    "url": "https://192.0.2.10/NIC_19.0.12_A00.EXE"
  }
]
}
```

Finally, launch the firmware update cleaning step against the node. The following example assumes the above JSON is in a file named `firmware_update.json`:

```
baremetal node clean <ironic_node_uuid> --clean-steps firmware_update.json
```

In the following example, the JSON is specified directly on the command line:

```
baremetal node clean <ironic_node_uuid> --clean-steps '[{"interface":
↪ "management", "step": "update_firmware", "args": {"firmware_images": [{"url
↪ ": "http://192.0.2.10/BMC_4_22_00_00.EXE", "wait": 300}, {"url": "https://
↪ 192.0.2.10/NIC_19.0.12_A00.EXE"}]}}]'
```

Note: Firmware updates may take some time to complete. If a firmware update cleaning step consistently times out, then consider performing fewer firmware updates in the cleaning step or increasing `clean_callback_timeout` in `ironic.conf` to increase the timeout value.

Warning: Warning: Removing power from a server while it is in the process of updating firmware may result in devices in the server, or the server itself becoming inoperable.

SNMP driver

The SNMP hardware type enables control of power distribution units of the type frequently found in data centre racks. PDUs frequently have a management ethernet interface and SNMP support enabling control of the power outlets.

The SNMP power interface works with the *PXE boot* interface for network deployment and network-configured boot.

Note: Unlike most of the other power interfaces, the SNMP power interface does not have a corresponding management interface. The SNMP hardware type uses the `noop` management interface instead.

List of supported devices

This is a non-exhaustive list of supported devices. Any device not listed in this table could possibly work using a similar driver.

Please report any device status.

Manufacturer	Model	Supported?	Driver name
APC	AP7920	Yes	apc_masterswitch
APC	AP9606	Yes	apc_masterswitch
APC	AP9225	Yes	apc_masterswitchplus
APC	AP7155	Yes	apc_rackpdu
APC	AP7900	Yes	apc_rackpdu
APC	AP7901	Yes	apc_rackpdu
APC	AP7902	Yes	apc_rackpdu
APC	AP7911a	Yes	apc_rackpdu
APC	AP7921	Yes	apc_rackpdu
APC	AP7922	Yes	apc_rackpdu
APC	AP7930	Yes	apc_rackpdu
APC	AP7931	Yes	apc_rackpdu
APC	AP7932	Yes	apc_rackpdu
APC	AP7940	Yes	apc_rackpdu
APC	AP7941	Yes	apc_rackpdu
APC	AP7951	Yes	apc_rackpdu
APC	AP7960	Yes	apc_rackpdu
APC	AP7990	Yes	apc_rackpdu
APC	AP7998	Yes	apc_rackpdu
APC	AP8941	Yes	apc_rackpdu
APC	AP8953	Yes	apc_rackpdu
APC	AP8959	Yes	apc_rackpdu
APC	AP8961	Yes	apc_rackpdu
APC	AP8965	Yes	apc_rackpdu
Aten	all?	Yes	aten
CyberPower	all?	Untested	cyberpower
EatonPower	all?	Untested	eatonpower
Teltronix	all?	Yes	teltronix
BayTech	MRP27	Yes	baytech_mrp27

Software Requirements

- The PySNMP package must be installed, variously referred to as `pysnmp` or `python-pysnmp`

Enabling the SNMP Hardware Type

1. Add `snmp` to the list of `enabled_hardware_types` in `ironic.conf`. Also update `enabled_management_interfaces` and `enabled_power_interfaces` in `ironic.conf` as shown below:

```
[DEFAULT]
enabled_hardware_types = snmp
enabled_management_interfaces = noop
enabled_power_interfaces = snmp
```

2. To set the default boot option, update `default_boot_option` in `ironic.conf`:

```
[DEFAULT]
default_boot_option = netboot
```

Note: Currently the default value of `default_boot_option` is `netboot` but it will be changed to `local` in the future. It is recommended to set an explicit value for this option.

Note: It is important to set `boot_option` to `netboot` as SNMP hardware type does not support setting of boot devices. One can also configure a node to boot using `netboot` by setting its `capabilities` and updating Nova flavor as described below:

```
baremetal node set --property capabilities="boot_option:netboot" <node>
openstack flavor set --property "capabilities:boot_option"="netboot"
↪ ironic-flavor
```

3. Restart the Ironic conductor service.

```
service ironic-conductor restart
```

Ironic Node Configuration

Nodes configured to use the SNMP hardware type should have the `driver` field set to the hardware type `snmp`.

The following property values have to be added to the nodes `driver_info` field:

- `snmp_driver`: PDU manufacturer driver name or `auto` to automatically choose ironic snmp driver based on `SNMPv2-MIB::sysObjectID` value as reported by PDU.
- `snmp_address`: the IPv4 address of the PDU controlling this node.

- `snmp_port`: (optional) A non-standard UDP port to use for SNMP operations. If not specified, the default port (161) is used.
- `snmp_outlet`: The power outlet on the PDU (1-based indexing).
- `snmp_version`: (optional) SNMP protocol version (permitted values 1, 2c or 3). If not specified, SNMPv1 is chosen.
- `snmp_community`: (Required for SNMPv1/SNMPv2c unless `snmp_community_read` and/or `snmp_community_write` properties are present in which case the latter take over) SNMP community name parameter for reads and writes to the PDU.
- `snmp_community_read`: SNMP community name parameter for reads to the PDU. Takes precedence over the `snmp_community` property.
- `snmp_community_write`: SNMP community name parameter for writes to the PDU. Takes precedence over the `snmp_community` property.
- `snmp_user`: (Required for SNMPv3) SNMPv3 User-based Security Model (USM) user name. Synonym for now obsolete `snmp_security` parameter.
- `snmp_auth_protocol`: SNMPv3 message authentication protocol ID. Valid values include: `none`, `md5`, `sha` for all `pysnmp` versions and additionally `sha224`, `sha256`, `sha384`, `sha512` for `pysnmp` versions 4.4.1 and later. Default is `none` unless `snmp_auth_key` is provided. In the latter case `md5` is the default.
- `snmp_auth_key`: SNMPv3 message authentication key. Must be 8+ characters long. Required when message authentication is used.
- `snmp_priv_protocol`: SNMPv3 message privacy (encryption) protocol ID. Valid values include: `none`, `des`, `3des`, `aes`, `aes192`, `aes256` for all `pysnmp` version and additionally `aes192blmt`, `aes256blmt` for `pysnmp` versions 4.4.3+. Note that message privacy requires using message authentication. Default is `none` unless `snmp_priv_key` is provided. In the latter case `des` is the default.
- `snmp_priv_key`: SNMPv3 message privacy (encryption) key. Must be 8+ characters long. Required when message encryption is used.
- `snmp_context_engine_id`: SNMPv3 context engine ID. Default is the value of authoritative engine ID.
- `snmp_context_name`: SNMPv3 context name. Default is an empty string.

The following command can be used to enroll a node with the `snmp` hardware type:

```
baremetal node create \
--driver snmp --driver-info snmp_driver=<pdu_manufacturer> \
--driver-info snmp_address=<ip_address> \
--driver-info snmp_outlet=<outlet_index> \
--driver-info snmp_community=<community_string> \
--properties capabilities=boot_option:netboot
```

XClarity driver

Overview

The `xclarity` driver is targeted for IMM 2.0 and IMM 3.0 managed Lenovo servers. The `xclarity` hardware type enables the user to take advantage of XClarity Manager by using the XClarity Python Client.

Prerequisites

- The XClarity Client library should be installed on the ironic conductor node(s).

For example, it can be installed with `pip`:

```
sudo pip install python-xclarityclient
```

Enabling the XClarity driver

1. Add `xclarity` to the list of `enabled_hardware_types`, `enabled_power_interfaces` and `enabled_management_interfaces` in `/etc/ironic/ironic.conf`. For example:

```
[DEFAULT]
...
enabled_hardware_types = ipmi,xclarity
enabled_power_interfaces = ipmitool,xclarity
enabled_management_interfaces = ipmitool,xclarity
```

2. Restart the ironic conductor service:

```
sudo service ironic-conductor restart

# Or, for RDO:
sudo systemctl restart openstack-ironic-conductor
```

Registering a node with the XClarity driver

Nodes configured to use the driver should have the `driver` property set to `xclarity`.

The following properties are specified in the nodes `driver_info` field and are required:

- `xclarity_manager_ip`: The IP address of the XClarity Controller.
- `xclarity_username`: User account with admin/server-profile access privilege to the XClarity Controller.
- `xclarity_password`: User account password corresponding to the `xclarity_username` to the XClarity Controller.
- `xclarity_hardware_id`: The hardware ID of the XClarity managed server.

The `baremetal node create` command can be used to enroll a node with the `xclarity` driver. For example:

```
baremetal node create --driver xclarity \
  --driver-info xclarity_manager_ip=https://10.240.217.101 \
  --driver-info xclarity_username=admin \
  --driver-info xclarity_password=password \
  --driver-info xclarity_hardware_id=hardware_id
```

For more information about enrolling nodes see *Enrollment* in the install guide.

4.1.3 Changing Hardware Types and Interfaces

Hardware types and interfaces are enabled in the configuration as described in *Enabling drivers and hardware types*. Usually, a hardware type is configured on enrolling as described in *Enrollment*:

```
baremetal node create --driver <hardware type>
```

Any hardware interfaces can be specified on enrollment as well:

```
baremetal node create --driver <hardware type> \
  --deploy-interface direct --<other>-interface <other implementation>
```

For the remaining interfaces the default value is assigned as described in *Defaults for hardware interfaces*. Both the hardware type and the hardware interfaces can be changed later via the node update API.

Changing Hardware Interfaces

Hardware interfaces can be changed by the following command:

```
baremetal node set <NODE> \
  --deploy-interface direct \
  --<other>-interface <other implementation>
```

The modified interfaces must be enabled and compatible with the current nodes hardware type.

Changing Hardware Type

Changing the nodes hardware type can pose a problem. When the `driver` field is updated, the final result must be consistent, that is, the resulting hardware interfaces must be compatible with the new hardware type. This will not work:

```
baremetal node create --name test --driver fake-hardware
baremetal node set test --driver ipmi
```

This is because the `fake-hardware` hardware type defaults to `fake` implementations for some or all interfaces, but the `ipmi` hardware type is not compatible with them. There are three ways to deal with this situation:

1. Provide new values for all incompatible interfaces, for example:

```
baremetal node set test --driver ipmi \  
  --boot-interface pxe \  
  --deploy-interface direct \  
  --management-interface ipmitool \  
  --power-interface ipmitool
```

2. Request resetting some of the interfaces to their new defaults by using the `--reset-<IFACE>-interface` family of arguments, for example:

```
baremetal node set test --driver ipmi \  
  --reset-boot-interface \  
  --reset-deploy-interface \  
  --reset-management-interface \  
  --reset-power-interface
```

Note: This feature is available starting with ironic 11.1.0 (Rocky series, API version 1.45).

3. Request resetting all interfaces to their new defaults:

```
baremetal node set test --driver ipmi --reset-interfaces
```

You can still specify explicit values for some interfaces:

```
baremetal node set test --driver ipmi --reset-interfaces \  
  --deploy-interface direct
```

Note: This feature is available starting with ironic 11.1.0 (Rocky series, API version 1.45).

Static boot order configuration

Some hardware is known to misbehave when changing the boot device through the BMC. To work around it you can use the noop management interface implementation with the `ipmi` and `redfish` hardware types. In this case the Bare Metal service will not change the boot device for you, leaving the pre-configured boot order.

For example, in case of the *PXE boot*:

1. Via any available means configure the boot order on the node as follows:
 1. Boot from PXE/iPXE on the provisioning NIC.

Warning: If it is not possible to limit network boot to only provisioning NIC, make sure that no other DHCP/PXE servers are accessible by the node.

2. Boot from hard drive.
2. Make sure the noop management interface is enabled, for example:


```
[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_management_interfaces = ipmitool,redfish,noop
```

3. Change the node to use the noop management interface:

```
baremetal node set <NODE> --management-interface noop
```

4.1.4 Unsupported drivers

The following drivers were declared as unsupported in ironic Newton release and as of Ocata release they are removed from ironic:

- AMT driver - available as part of [ironic-staging-drivers](#)
- iBoot driver - available as part of [ironic-staging-drivers](#)
- Wake-On-Lan driver - available as part of [ironic-staging-drivers](#)
- Virtualbox drivers
- SeaMicro drivers
- MSFT OCS drivers

The SSH drivers were removed in the Pike release. Similar functionality can be achieved either with [VirtualBMC](#) or using libvirt drivers from [ironic-staging-drivers](#).

4.2 Administrators Guide

If you are a system administrator running Ironic, this section contains information that may help you understand how to operate and upgrade the services.

4.2.1 Ironic Python Agent

Overview

Ironic Python Agent (also often called *IPA* or just *agent*) is a Python-based agent which handles *ironic* bare metal nodes in a variety of actions such as inspect, configure, clean and deploy images. IPA is distributed over nodes and runs, inside of a ramdisk, the process of booting this ramdisk on the node.

For more information see the [ironic-python-agent documentation](#).

Drivers

Starting with the Kilo release all deploy interfaces (except for fake ones) are using IPA. For nodes using the *Direct deploy* interface, the conductor prepares a swift temporary URL or a local HTTP URL for the image. IPA then handles the whole deployment process: downloading an image from swift, putting it on the machine and doing any post-deploy actions.

Requirements

Using IPA requires it to be present and configured on the deploy ramdisk, see *Building or downloading a deploy ramdisk image*

Using proxies for image download

Overview

When using the *Direct deploy*, IPA supports using proxies for downloading the user image. For example, this could be used to speed up download by using a caching proxy.

Steps to enable proxies

1. Configure the proxy server of your choice (for example [Squid](#), [Apache Traffic Server](#)). This will probably require you to configure the proxy server to cache the content even if the requested URL contains a query, and to raise the maximum cached file size as images can be pretty big. If you have HTTPS enabled in swift (see [swift deployment guide](#)), it is possible to configure the proxy server to talk to swift via HTTPS to download the image, store it in the cache unencrypted and return it to the node via HTTPS again. Because the image will be stored unencrypted in the cache, this approach is recommended for images that do not contain sensitive information. Refer to your proxy servers documentation to complete this step.
2. Set `[glance]swift_temp_url_cache_enabled` in the ironic conductor config file to `True`. The conductor will reuse the cached swift temporary URLs instead of generating new ones each time an image is requested, so that the proxy server does not create new cache entries for the same image, based on the query part of the URL (as it contains some query parameters that change each time it is regenerated).
3. Set `[glance]swift_temp_url_expected_download_start_delay` option in the ironic conductor config file to the value appropriate for your hardware. This is the delay (in seconds) from the time of the deploy request (when the swift temporary URL is generated) to when the URL is used for the image download. You can think of it as roughly the time needed for IPA ramdisk to startup and begin download. This value is used to check if the swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled, this will determine if a cached entry will still be valid when the download starts. It is used only if `[glance]swift_temp_url_cache_enabled` is `True`.
4. Increase `[glance]swift_temp_url_duration` option in the ironic conductor config file, as only non-expired links to images will be returned from the swift temporary URLs cache. This means that if `swift_temp_url_duration=1200` then after 20 minutes a new image will be cached by the proxy server as the query in its URL will change. The value of this option must be greater than or equal to `[glance]swift_temp_url_expected_download_start_delay`.

5. Add one or more of `image_http_proxy`, `image_https_proxy`, `image_no_proxy` to `driver_info` properties in each node that will use the proxy.

Advanced configuration

Out-of-band vs. in-band power off on deploy

After deploying an image onto the nodes hard disk, Ironic will reboot the machine into the new image. By default this power action happens in-band, meaning that the ironic-conductor will instruct the IPA ramdisk to power itself off.

Some hardware may have a problem with the default approach and would require Ironic to talk directly to the management controller to switch the power off and on again. In order to tell Ironic to do that, you have to update the nodes `driver_info` field and set the `deploy_forces_oob_reboot` parameter with the value of **True**. For example, the below command sets this configuration in a specific node:

```
baremetal node set <UUID or name> --driver-info deploy_forces_oob_reboot=True
```

4.2.2 Hardware Inspection

Overview

Inspection allows Bare Metal service to discover required node properties once required `driver_info` fields (for example, IPMI credentials) are set by an operator. Inspection will also create the Bare Metal service ports for the discovered ethernet MACs. Operators will have to manually delete the Bare Metal service ports for which physical media is not connected. This is required due to the [bug 1405131](#).

There are two kinds of inspection supported by Bare Metal service:

1. Out-of-band inspection is currently implemented by several hardware types, including `ilo`, `idrac` and `irmc`.
2. *In-band inspection* by utilizing the `ironic-inspector` project.

The node should be in the `manageable` state before inspection is initiated. If it is in the `enroll` or `available` state, move it to `manageable` first:

```
baremetal node manage <node_UUID>
```

Then inspection can be initiated using the following command:

```
baremetal node inspect <node_UUID>
```

Capabilities discovery

This is an incomplete list of capabilities we want to discover during inspection. The exact support is hardware and hardware type specific though, the most complete list is provided by the iLO *Hardware Inspection Support*.

secure_boot (true or false) whether secure boot is supported for the node

boot_mode (bios or uefi) the boot mode the node is using

cpu_vt (true or false) whether the CPU virtualization is enabled

cpu_aes (true or false) whether the AES CPU extensions are enabled

max_raid_level (integer, 0-10) maximum RAID level supported by the node

pci_gpu_devices (non-negative integer) number of GPU devices on the node

The operator can specify these capabilities in nova flavor for node to be selected for scheduling:

```
nova flavor-key my-baremetal-flavor set capabilities:pci_gpu_devices="> 0"  
nova flavor-key my-baremetal-flavor set capabilities:secure_boot="true"
```

Please see a specific *hardware type page* for the exact list of capabilities this hardware type can discover.

In-band inspection

In-band inspection involves booting a ramdisk on the target node and fetching information directly from it. This process is more fragile and time-consuming than the out-of-band inspection, but it is not vendor-specific and works across a wide range of hardware. In-band inspection is using the *ironic-inspector* project.

It is supported by all hardware types, and used by default, if enabled, by the *ipmi* hardware type. The *inspector inspect* interface has to be enabled to use it:

```
[DEFAULT]  
enabled_inspect_interfaces = inspector,no-inspect
```

If the *ironic-inspector* service is not registered in the service catalog, set the following option:

```
[inspector]  
endpoint_override = http://inspector.example.com:5050
```

In order to ensure that ports in Bare Metal service are synchronized with NIC ports on the node, the following settings in the *ironic-inspector* configuration file must be set:

```
[processing]  
add_ports = all  
keep_ports = present
```

4.2.3 Node Deployment

Contents

- *Node Deployment*
 - *Overview*
 - *Deploy Steps*
 - *Deploy Templates*

Overview

Node deployment is performed by the Bare Metal service to prepare a node for use by a workload. The exact work flow used depends on a number of factors, including the hardware type and interfaces assigned to a node.

Deploy Steps

The Bare Metal service implements deployment by collecting a list of deploy steps to perform on a node from the Power, Deploy, Management, BIOS, and RAID interfaces of the driver assigned to the node. These steps are then ordered by priority and executed on the node when the node is moved to the deploying state.

Nodes move to the `deploying` state when attempting to move to the `active` state (when the hardware is prepared for use by a workload). For a full understanding of all state transitions into deployment, please see *Ironics State Machine*.

The Bare Metal service added support for deploy steps in the Rocky release.

Order of execution

Deploy steps are ordered from higher to lower priority, where a larger integer is a higher priority. If the same priority is used by deploy steps on different interfaces, the following resolution order is used: Power, Management, Deploy, BIOS, and RAID interfaces.

Agent steps

All deploy interfaces based on `ironic-python-agent` (i.e. `direct`, `ansible` and any derivatives) expose the following deploy steps:

`deploy.deploy` (priority 100) In this step the node is booted using a provisioning image.

`deploy.write_image` (priority 80) An out-of-band (`ansible`) or in-band (`direct`) step that downloads and writes the image to the node.

`deploy.tear_down_agent` (priority 40) In this step the provisioning image is shut down.

`deploy.switch_to_tenant_network` (priority 30) In this step networking for the node is switched from provisioning to tenant networks.

deploy.boot_instance (priority 20) In this step the node is booted into the user image.

Additionally, the direct deploy interfaces has:

deploy.prepare_instance_boot (priority 60) In this step the boot device is configured and the bootloader is installed.

Note: For the `ansible` deploy interface these steps are done in `deploy.write_image`.

Accordingly, the following priority ranges can be used for custom deploy steps:

> **100** Out-of-band steps to run before deployment.

81 to 99 In-band deploy steps to run before the image is written.

61 to 79 In-band deploy steps to run after the image is written but before the bootloader is installed.

41 to 59 In-band steps to run after the image is written the bootloader is installed.

21 to 39 Out-of-band steps to run after the provisioning image is shut down.

1 to 19 Any steps that are run when the user instance is already running.

In-band steps

More deploy steps can be provided by the ramdisk, see [IPA hardware managers documentation](#) for a listing.

Requesting steps

Starting with Bare Metal API version 1.69 user can optionally supply deploy steps for node deployment when invoking deployment or rebuilding. Overlapping steps will take precedence over *Agent steps* and *Deploy Templates* steps.

Using baremetal client deploy steps can be passed via `--deploy-steps` argument. The argument `--deploy-steps` is one of:

- a JSON string
- path to a JSON file whose contents are passed to the API
- -, to read from stdin. This allows piping in the deploy steps.

An example by passing a JSON string:

```
baremetal node deploy <node> \  
  --deploy-steps '[{"interface": "bios", "step": "apply_configuration",  
↪ "args": {"settings": [{"name": "LogicalProc", "value": "Enabled"}]},  
↪ "priority": 150}]'
```

Format of JSON for deploy steps argument is described in *Deploy step format* section.

Note: Starting with *ironicclient* 4.6.0 you can provide a YAML file for `--deploy-steps`.

Excluding the default steps

Starting with the Xena release, you can use the new *Custom agent deploy* interface to exclude the default step `write_image` and skip bootloader installation in the `prepare_instance_boot` step.

Writing a Deploy Step

Please refer to *Developing a new Deploy Step*.

FAQ

What deploy step is running?

To check what deploy step the node is performing or attempted to perform and failed, run the following command; it will return the value in the nodes `driver_internal_info` field:

```
baremetal node show <node> -f value -c driver_internal_info
```

The `deploy_steps` field will contain a list of all remaining steps with their priorities, and the first one listed is the step currently in progress or that the node failed before going into `deploy failed` state.

Troubleshooting

If deployment fails on a node, the node will be put into the `deploy failed` state until the node is deprovisioned. A deprovisioned node is moved to the `available` state after the cleaning process has been performed successfully.

Strategies for determining why a deploy step failed include checking the ironic conductor logs, checking logs from the ironic-python-agent that have been stored on the ironic conductor, or performing general hardware troubleshooting on the node.

Deploy Templates

Starting with the Stein release, with Bare Metal API version 1.55, deploy templates offer a way to define a set of one or more deploy steps to be executed with particular sets of arguments and priorities.

Each deploy template has a name, which must be a valid trait. Traits can be either standard or custom. Standard traits are listed in the `os_traits` library. Custom traits must meet the following requirements:

- prefixed with `CUSTOM_`
- contain only upper case characters A to Z, digits 0 to 9, or underscores
- no longer than 255 characters in length

Deploy step format

An invocation of a deploy step is defined in a deploy template as follows:

```
{
  "interface": "<name of the driver interface>",
  "step": "<name of the step>",
  "args": {
    "<arg1>": "<value1>",
    "<arg2>": "<value2>"
  },
  "priority": <priority of the step>
}
```

A deploy template contains a list of one or more such steps. Each combination of *interface* and *step* may only be specified once in a deploy template.

Matching deploy templates

During deployment, if any of the traits in a nodes `instance_info.traits` field match the name of a deploy template, then the steps from that deploy template will be added to the list of steps to be executed by the node.

When using the Compute service, any traits in the instances flavor properties or image properties are stored in `instance_info.traits` during deployment. See [Scheduling based on traits](#) for further information on how traits are used for scheduling when the Bare Metal service is used with the Compute service.

Note that there is no ongoing relationship between a node and any templates that are matched during deployment. The set of matching deploy templates is checked at deployment time. Any subsequent updates to or deletion of those templates will not be reflected in the nodes configuration unless it is redeployed or rebuilt. Similarly, if a node is rebuilt and the set of matching deploy templates has changed since the initial deployment, then the resulting configuration of the node may be different from the initial deployment.

Overriding default deploy steps

A deploy step is enabled by default if it has a non-zero default priority. A default deploy step may be overridden in a deploy template. If the steps priority is a positive integer it will be executed with the specified priority and arguments. If the steps priority is zero, the step will not be executed.

If a [core deploy step](#) is included in a deploy template, it can only be assigned a priority of zero to disable it.

Creating a deploy template via API

A deploy template can be created using the Bare Metal API:

```
POST /v1/deploy_templates
```

Here is an example of the body of a request to create a deploy template with a single step:

```
{
  "name": "CUSTOM_HYPERTHREADING_ON",
  "steps": [
    {
      "interface": "bios",
      "step": "apply_configuration",
      "args": {
        "settings": [
          {
            "name": "LogicalProc",
            "value": "Enabled"
          }
        ]
      },
      "priority": 150
    }
  ]
}
```

Further information on this API is available [here](#).

Creating a deploy template via baremetal client

A deploy template can be created via the `baremetal deploy template create` command, starting with `python-ironicclient 2.7.0`.

The argument `--steps` must be specified. Its value is one of:

- a JSON string
- path to a JSON file whose contents are passed to the API
- -, to read from stdin. This allows piping in the deploy steps.

Example of creating a deploy template with a single step using a JSON string:

```
baremetal deploy template create \
  CUSTOM_HYPERTHREADING_ON \
  --steps '[{"interface": "bios", "step": "apply_configuration", "args": {
↵"settings": [{"name": "LogicalProc", "value": "Enabled"}]}, "priority": 150}
↵]'
```

Or with a file:

```
baremetal deploy template create \  
  CUSTOM_HYPERTHREADING_ON \  
  ---steps my-deploy-steps.txt
```

Or with stdin:

```
cat my-deploy-steps.txt | baremetal deploy template create \  
  CUSTOM_HYPERTHREADING_ON \  
  --steps -
```

Example of use with the Compute service

Note: The deploy steps used in this example are for example purposes only.

In the following example, we first add the trait `CUSTOM_HYPERTHREADING_ON` to the node represented by `<node>`:

```
baremetal node add trait <node> CUSTOM_HYPERTHREADING_ON
```

We also update the flavor `bm-hyperthreading-on` in the Compute service with the following property:

```
openstack flavor set --property trait:CUSTOM_HYPERTHREADING_ON=required bm-  
↪hyperthreading-on
```

Creating a Compute instance with this flavor will ensure that the instance is scheduled only to Bare Metal nodes with the `CUSTOM_HYPERTHREADING_ON` trait.

We could then create a Bare Metal deploy template with the name `CUSTOM_HYPERTHREADING_ON` and a deploy step that enables Hyperthreading:

```
{  
  "name": "CUSTOM_HYPERTHREADING_ON",  
  "steps": [  
    {  
      "interface": "bios",  
      "step": "apply_configuration",  
      "args": {  
        "settings": [  
          {  
            "name": "LogicalProc",  
            "value": "Enabled"  
          }  
        ]  
      },  
      "priority": 150  
    }  
  ]  
}
```

When an instance is created using the `bm-hypervthreading-on` flavor, then the deploy steps of deploy template `CUSTOM_HYPERTHREADING_ON` will be executed during the deployment of the scheduled node, causing Hyperthreading to be enabled in the nodes BIOS configuration.

To make this example more dynamic, lets add a second trait `CUSTOM_HYPERTHREADING_OFF` to the node:

```
baremetal node add trait <node> CUSTOM_HYPERTHREADING_OFF
```

We could also update a second flavor, `bm-hypervthreading-off`, with the following property:

```
openstack flavor set --property trait:CUSTOM_HYPERTHREADING_OFF=required bm-
↪hypervthreading-off
```

Finally, we create a deploy template with the name `CUSTOM_HYPERTHREADING_OFF` and a deploy step that disables Hyperthreading:

```
{
  "name": "CUSTOM_HYPERTHREADING_OFF",
  "steps": [
    {
      "interface": "bios",
      "step": "apply_configuration",
      "args": {
        "settings": [
          {
            "name": "LogicalProc",
            "value": "Disabled"
          }
        ]
      }
    },
    {
      "priority": 150
    }
  ]
}
```

Creating a Compute instance with the `bm-hypervthreading-off` instance will cause the scheduled node to have Hyperthreading disabled in the BIOS during deployment.

We now have a way to create Compute instances with different configurations, by choosing between different Compute flavors, supported by a single Bare Metal node that is dynamically configured during deployment.

4.2.4 Node cleaning

Overview

Ironic provides two modes for node cleaning: `automated` and `manual`.

`Automated` cleaning is automatically performed before the first workload has been assigned to a node and when hardware is recycled from one workload to another.

`Manual` cleaning must be invoked by the operator.

Automated cleaning

When hardware is recycled from one workload to another, ironic performs automated cleaning on the node to ensure its ready for another workload. This ensures the tenant will get a consistent bare metal node deployed every time.

Ironic implements automated cleaning by collecting a list of cleaning steps to perform on a node from the Power, Deploy, Management, BIOS, and RAID interfaces of the driver assigned to the node. These steps are then ordered by priority and executed on the node when the node is moved to cleaning state, if automated cleaning is enabled.

With automated cleaning, nodes move to cleaning state when moving from active -> available state (when the hardware is recycled from one workload to another). Nodes also traverse cleaning when going from manageable -> available state (before the first workload is assigned to the nodes). For a full understanding of all state transitions into cleaning, please see *Ironics State Machine*.

Ironic added support for automated cleaning in the Kilo release.

Enabling automated cleaning

To enable automated cleaning, ensure that your ironic.conf is set as follows:

```
[conductor]
automated_clean=true
```

This will enable the default set of cleaning steps, based on your hardware and ironic hardware types used for nodes. This includes, by default, erasing all of the previous tenants data.

You may also need to configure a *Cleaning Network*.

Cleaning steps

Cleaning steps used for automated cleaning are ordered from higher to lower priority, where a larger integer is a higher priority. In case of a conflict between priorities across interfaces, the following resolution order is used: Power, Management, Deploy, BIOS, and RAID interfaces.

You can skip a cleaning step by setting the priority for that cleaning step to zero or None.

You can reorder the cleaning steps by modifying the integer priorities of the cleaning steps.

See *How do I change the priority of a cleaning step?* for more information.

Management Interface

Table 1: idrac cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
clear_job_queue	Clear the job queue.	0	no	
known_good_state	Reset the iDRAC, Clear the job queue.	0	no	
reset_idrac	Reset the iDRAC.	0	no	

Table 2: idrac-redfish cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
clear_job_queue	Clear iDRAC job queue.	0	no	
clear_secure_boot_keys	Clears secure boot keys.	0	no	
export_configuration	<p>Export the configuration of the server.</p> <p>Exports the configuration of the server against which the step is run and stores it in specific format in indicated location.</p> <p>Uses Dells Server Configuration Profile (SCP) from <i>sushy-oem-idrac</i> library to get ALL configuration for cloning.</p> <p>param task A task from TaskManager.</p> <p>param export_configuration_location URL of location to save the configuration to.</p>	0	no	export_configuration_location (<i>required</i>) URL of location to save the configuration to.
import_configuration	<p>Import and apply the configuration to the server.</p> <p>Gets pre-created configuration from storage by given location and imports that into given server. Uses Dells Server Configuration Profile (SCP).</p> <p>param task A task from TaskManager.</p> <p>param import_configuration_location URL of location to fetch desired configuration from.</p>	0	no	import_configuration_location (<i>required</i>) URL of location to fetch desired configuration from.
import_export_configuration	<p>Import and export configuration in one go.</p> <p>Gets pre-created configuration from storage by given name and imports that into given server. After that ex-</p>	0	no	export_configuration_location (<i>required</i>) URL of location to save the configuration to. import_configuration_location (<i>required</i>) URL of location to fetch desired configura-

Table 3: idrac-wsman cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
clear_job_queue	Clear the job queue.	0	no	
known_good_state	Reset the iDRAC, Clear the job queue.	0	no	
reset_idrac	Reset the iDRAC.	0	no	

Table 4: ilo cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
activate_license	Activates iLO Advanced license.	0	no	<code>ilo_license_key</code> (<i>required</i>) The HPE iLO Advanced license key to activate enterprise features.
clear_secure_boot_keys	Clears all secure boot keys. Clears all the secure boot keys. This operation is supported only on HP Proliant Gen9 and above servers.	0	no	
reset_bios_to_defaults	Resets the BIOS settings to default values. Resets BIOS to default settings. This operation is currently supported only on HP Proliant Gen9 and above servers.	10	no	
reset_ilo	Resets the iLO.	0	no	
reset_ilo_credentials	Resets the iLO password.	30	no	
reset_secure_boot_defaults	Resets secure boot keys to manufacturing defaults. Resets the secure boot keys to manufacturing defaults. This operation is supported only on HP Proliant Gen9 and above servers.	20	no	
security_parameters_update	Updates the security parameters.	0	no	<code>security_parameters</code> (<i>required</i>) This argument represents the ordered list of JSON dictionaries of security parameters. Each security parameter consists of three fields, namely <code>param</code> , <code>ignore</code> and <code>enable</code> from which <code>param</code> field will be mandatory. These fields represent security parameter name, ignore flag and state of the security parameter. The supported security parameter names are <code>password_complexity</code> , <code>require_login_for_ilo_rbsu</code> , <code>ipmi_over_lan</code> , <code>secure_boot</code> , <code>require_host_authentication</code> . The security parameters will be updated (in the order given) one by one on the baremetal server.
update_auth_fail_logging_threshold	Updates the logging threshold security parameter.	0	no	<code>ignore</code> This argument represents boolean parameter. If set True the security parameters will be ignored by

Table 5: ilo5 cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
activate_license	Activates iLO Advanced license.	0	no	ilo_license_key (required) The HPE iLO Advanced license key to activate enterprise features.
clear_secure_boot	Clears all secure boot keys. Clears all the secure boot keys. This operation is supported only on HP Proliant Gen9 and above servers.	0	no	
erase_devices	Erase all the drives on the node. This method performs out-of-band sanitize disk erase on all the supported physical drives in the node. This erase cannot be performed on logical drives.	0	no	erase_pattern Dictionary of disk type and corresponding erase pattern to be used to perform specific out-of-band sanitize disk erase. Supported values are, for hdd: (overwrite, crypto, zero), for ssd: (block, crypto, zero). Default pattern is: {hdd: overwrite, ssd: block}.
one_button_secure_erase	Erase the whole system securely. The One-button secure erase process resets iLO and deletes all licenses stored there, resets BIOS settings, and deletes all Active Health System (AHS) and warranty data stored on the system. It also erases supported non-volatile storage data and deletes any deployment setting profiles.	0	no	
reset_bios_to_defaults	Reset the BIOS settings to default values. Resets BIOS to default settings. This operation is currently supported only on HP Proliant Gen9 and above servers.	10	no	
reset_ilo	Resets the iLO.	0	no	
reset_ilo_credentials	Resets the iLO password.	30	no	
reset_secure_boot	Resets secure boot keys to manufacturing defaults. Resets the secure boot keys to manufacturing defaults. This operation is supported only on HP Proliant Gen9 and above servers.	20	no	
update_security_parameters	Update the security parameters.	0	no	security_parameters (required) This argument represents the ordered list of JSON dictionaries of

Table 6: irmc cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
restore_irmc_bios	Restores BIOS config for a node.	0	no	

Table 7: redfish cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
clear_secure_boot_keys	Clears secure boot keys.	0	no	
reset_secure_boot_keys	Resets secure boot keys to manufacturing defaults.	0	no	
update_firmware	Updates the firmware on the node.	0	no	firmware_images (<i>required</i>) A list of firmware images to apply.

Bios Interface

Table 8: idrac-redfish cleaning steps

Name	Details	Prior-ity	Stop-pable	Arguments
apply_configuration	Apply the BIOS settings to the node.	0	no	settings (<i>required</i>) A list of BIOS settings to be applied
factory_reset	Reset the BIOS settings of the node to the factory default.	0	no	

Table 9: idrac-wsman cleaning steps

Name	Details	Priority	Stoppable	Arguments
apply_configuration	<p>Apply the BIOS configuration to the node</p> <p>param task a TaskManager instance containing the node to act on</p> <p>param settings List of BIOS settings to apply</p> <p>raises DRACOperationError upon an error from python-dracclient</p>	0	no	settings (<i>required</i>) List of BIOS settings to apply
factory_reset	<p>Reset the BIOS settings of the node to the factory default.</p> <p>This uses the Lifecycle Controller configuration to perform BIOS configuration reset. Leveraging the python-dracclient methods already available.</p>	0	no	

Table 10: ilo cleaning steps

Name	Details	Priority	Stoppable	Arguments
apply_configuration	<p>Applies the provided configuration on the node.</p>	0	no	settings (<i>required</i>) Dictionary with current BIOS configuration.
factory_reset	<p>Reset the BIOS settings to factory configuration.</p>	0	no	

Table 11: irmc cleaning steps

Name	Details	Priority	Stoppable	Arguments
apply_configuration	<p>Applies BIOS configuration on the given node.</p> <p>This method takes the BIOS settings from the settings param and applies BIOS configuration on the given node. After the BIOS configuration is done, self.cache_bios_settings() may be called to sync the nodes BIOS-related information with the BIOS configuration applied on the node. It will also validate the given settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties.</p>	0	no	settings (<i>required</i>) Dictionary containing the BIOS configuration.

Table 12: redfish cleaning steps

Name	Details	Priority	Stoppable	Arguments
apply_configuration	Apply the BIOS settings to the node.	0	no	settings (<i>required</i>) A list of BIOS settings to be applied
factory_reset	Reset the BIOS settings of the node to the factory default.	0	no	

Raid Interface

Table 13: agent cleaning steps

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create a RAID configuration on a bare metal using agent ramdisk. This method creates a RAID configuration on the given node.	0	no	
delete_configuration	Deletes RAID configuration on the given node.	0	no	

Table 14: ibmc cleaning steps

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create a RAID configuration. This method creates a RAID configuration on the given node.	0	no	<p>create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p>create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i>.</p> <p>delete_existing Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i>.</p>
delete_configuration	Deletes the RAID configuration.	0	no	

Table 15: idrac cleaning steps

Name	Details	Priority	Stop-pable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	<code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i> . <code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i> . <code>delete_existing</code> Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i> .
delete_configuration	Delete the RAID configuration.	0	no	

Table 16: idrac-redfish cleaning steps

Name	Details	Priority	Stop-pable	Arguments
create_configuration	Create RAID configuration on the node. This method creates the RAID configuration as read from <code>node.target_raid_config</code> . This method by default will create all logical disks.	0	no	<code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i> . <code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i> . <code>delete_existing</code> Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i> .
delete_configuration	Delete RAID configuration on the node.	0	no	

Table 17: idrac-wsman cleaning steps

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	<p><code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p><code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i>.</p> <p><code>delete_existing</code> Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i>.</p>
delete_configuration	Delete the RAID configuration.	0	no	

Table 18: ilo5 cleaning steps

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create a RAID configuration on a bare metal using agent ramdisk. This method creates a RAID configuration on the given node.	0	no	<p><code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p><code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i>.</p>
delete_configuration	Delete the RAID configuration.	0	no	

Table 19: irmc cleaning steps

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	<p><code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p><code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i>.</p>
delete_configuration	Delete the RAID configuration.	0	no	

Table 20: redfish cleaning steps

Name	Details	Priority	Stop-pable	Arguments
create_configuration	<p>Creates RAID configuration on the node.</p> <p>This method creates the RAID configuration as read from node.target_raid_config. This method by default will create all logical disks.</p>	0	no	<p><code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p><code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i>.</p> <p><code>delete_existing</code> Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i>.</p>
delete_configuration	<p>Deletes RAID configuration on the node.</p>	0	no	

Manual cleaning

Manual cleaning is typically used to handle long running, manual, or destructive tasks that an operator wishes to perform either before the first workload has been assigned to a node or between workloads. When initiating a manual clean, the operator specifies the cleaning steps to be performed. Manual cleaning can only be performed when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again.

Ironic added support for manual cleaning in the 4.4 (Mitaka series) release.

Setup

In order for manual cleaning to work, you may need to configure a *Cleaning Network*.

Starting manual cleaning via API

Manual cleaning can only be performed when a node is in the `manageable` state. The REST API request to initiate it is available in API version 1.15 and higher:

```
PUT /v1/nodes/<node_ident>/states/provision
```

(Additional information is available [here](#).)

This API will allow operators to put a node directly into `cleaning provision` state from `manageable` state via `target: clean`. The PUT will also require the argument `clean_steps` to be specified. This is an ordered list of cleaning steps. A cleaning step is represented by a dictionary (JSON), in the form:

```
{
  "interface": "<interface>",
  "step": "<name of cleaning step>",
  "args": {"<arg1>": "<value1>", ..., "<argn>": <valuen>}
}
```

The interface and step keys are required for all steps. If a cleaning step method takes keyword arguments, the args key may be specified. It is a dictionary of keyword variable arguments, with each keyword-argument entry being <name>: <value>.

If any step is missing a required keyword argument, manual cleaning will not be performed and the node will be put in `clean failed` provision state with an appropriate error message.

If, during the cleaning process, a cleaning step determines that it has incorrect keyword arguments, all earlier steps will be performed and then the node will be put in `clean failed` provision state with an appropriate error message.

An example of the request body for this API:

```
{
  "target": "clean",
  "clean_steps": [{
    "interface": "raid",
    "step": "create_configuration",
    "args": {"create_nonroot_volumes": false}
  },
  {
    "interface": "deploy",
    "step": "erase_devices"
  }
  ]
}
```

In the above example, the nodes RAID interface would configure hardware RAID without non-root volumes, and then all devices would be erased (in that order).

Starting manual cleaning via openstack metal CLI

Manual cleaning is available via the `baremetal node clean` command, starting with Bare Metal API version 1.15.

The argument `--clean-steps` must be specified. Its value is one of:

- a JSON string
- path to a JSON file whose contents are passed to the API
- -, to read from stdin. This allows piping in the clean steps. Using - to signify stdin is common in Unix utilities.

The following examples assume that the Bare Metal API version was set via the `OS_BAREMETAL_API_VERSION` environment variable. (The alternative is to add `--os-baremetal-api-version 1.15` to the command.):


```
export OS_BAREMETAL_API_VERSION=1.15
```

Examples of doing this with a JSON string:

```
baremetal node clean <node> \
  --clean-steps '[{"interface": "deploy", "step": "erase_devices_metadata"}]
↪'

baremetal node clean <node> \
  --clean-steps '[{"interface": "deploy", "step": "erase_devices"}]'
```

Or with a file:

```
baremetal node clean <node> \
  --clean-steps my-clean-steps.txt
```

Or with stdin:

```
cat my-clean-steps.txt | baremetal node clean <node> \
  --clean-steps -
```

Cleaning Network

If you are using the Neutron DHCP provider (the default) you will also need to ensure you have configured a cleaning network. This network will be used to boot the ramdisk for in-band cleaning. You can use the same network as your tenant network. For steps to set up the cleaning network, please see *Configure the Bare Metal service for cleaning*.

In-band vs out-of-band

Ironic uses two main methods to perform actions on a node: in-band and out-of-band. Ironic supports using both methods to clean a node.

In-band

In-band steps are performed by ironic making API calls to a ramdisk running on the node using a deploy interface. Currently, all the deploy interfaces support in-band cleaning. By default, ironic-python-agent ships with a minimal cleaning configuration, only erasing disks. However, you can add your own cleaning steps and/or override default cleaning steps with a custom Hardware Manager.

Out-of-band

Out-of-band are actions performed by your management controller, such as IPMI, iLO, or DRAC. Out-of-band steps will be performed by ironic using a power or management interface. Which steps are performed depends on the hardware type and hardware itself.

For Out-of-Band cleaning operations supported by iLO hardware types, refer to *Node Cleaning Support*.

FAQ

How are cleaning steps ordered?

For automated cleaning, cleaning steps are ordered by integer priority, where a larger integer is a higher priority. In case of a conflict between priorities across hardware interfaces, the following resolution order is used:

1. Power interface
2. Management interface
3. Deploy interface
4. BIOS interface
5. RAID interface

For manual cleaning, the cleaning steps should be specified in the desired order.

How do I skip a cleaning step?

For automated cleaning, cleaning steps with a priority of 0 or None are skipped.

How do I change the priority of a cleaning step?

For manual cleaning, specify the cleaning steps in the desired order.

For automated cleaning, it depends on whether the cleaning steps are out-of-band or in-band.

Most out-of-band cleaning steps have an explicit configuration option for priority.

Changing the priority of an in-band (ironic-python-agent) cleaning step requires use of a custom HardwareManager. The only exception is `erase_devices`, which can have its priority set in `ironic.conf`. For instance, to disable `erase_devices`, you'd set the following configuration option:

```
[deploy]
erase_devices_priority=0
```

To enable/disable the in-band disk erase using iLO hardware type, use the following configuration option:

```
[ilo]
clean_priority_erase_devices=0
```

The generic hardware manager first identifies whether a device is an NVMe drive or an ATA drive so that it can attempt a platform-specific secure erase method. In case of NVMe drives, it tries to perform a secure format operation by using the `nvme-cli` utility. This behavior can be controlled using the following configuration option (by default it is set to `True`):

```
[deploy]
enable_nvme_secure_erase=True
```

In case of ATA drives, it tries to perform ATA disk erase by using the `hdparm` utility.

If neither method is supported, it performs software based disk erase using the `shred` utility. By default, the number of iterations performed by `shred` for software based disk erase is 1. To configure the number of iterations, use the following configuration option:

```
[deploy]
erase_devices_iterations=1
```

Overriding step priority

`[conductor]clean_step_priority_override` is a new configuration option which allows specifying priority of each step using multiple configuration values:

```
[conductor]
clean_step_priority_override=deploy.erase_devices_metadata:123
clean_step_priority_override=management.reset_bios_to_default:234
clean_step_priority_override=management.clean_priority_reset_ilo:345
```

This parameter can be specified as many times as required to define priorities for several cleaning steps - the values will be combined.

What cleaning step is running?

To check what cleaning step the node is performing or attempted to perform and failed, run the following command; it will return the value in the nodes `driver_internal_info` field:

```
baremetal node show $node_ident -f value -c driver_internal_info
```

The `clean_steps` field will contain a list of all remaining steps with their priorities, and the first one listed is the step currently in progress or that the node failed before going into `clean failed` state.

Should I disable automated cleaning?

Automated cleaning is recommended for ironic deployments, however, there are some tradeoffs to having it enabled. For instance, ironic cannot deploy a new instance to a node that is currently cleaning, and cleaning can be a time consuming process. To mitigate this, we suggest using NVMe drives with support for NVMe Secure Erase (based on `nvme-cli` format command) or ATA drives with support for cryptographic ATA Security Erase, as typically the `erase_devices` step in the deploy interface takes the longest time to complete of all cleaning steps.

Why cant I power on/off a node while its cleaning?

During cleaning, nodes may be performing actions that shouldnt be interrupted, such as BIOS or Firmware updates. As a result, operators are forbidden from changing power state via the ironic API while a node is cleaning.

Troubleshooting

If cleaning fails on a node, the node will be put into `clean failed` state. If the failure happens while running a clean step, the node is also placed in maintenance mode to prevent ironic from taking actions on the node. The operator should validate that no permanent damage has been done to the node and no processes are still running on it before removing the maintenance mode.

Note: Older versions of ironic may put the node to maintenance even when no clean step has been running.

Nodes in `clean failed` will not be powered off, as the node might be in a state such that powering it off could damage the node or remove useful information about the nature of the cleaning failure.

A `clean failed` node can be moved to `manageable` state, where it cannot be scheduled by nova and you can safely attempt to fix the node. To move a node from `clean failed` to `manageable`:

```
baremetal node manage $node_ident
```

You can now take actions on the node, such as replacing a bad disk drive.

Strategies for determining why a cleaning step failed include checking the ironic conductor logs, viewing logs on the still-running ironic-python-agent (if an in-band step failed), or performing general hardware troubleshooting on the node.

When the node is repaired, you can move the node back to `available` state, to allow it to be scheduled by nova.

```
# First, move it out of maintenance mode
baremetal node maintenance unset $node_ident

# Now, make the node available for scheduling by nova
baremetal node provide $node_ident
```

The node will begin automated cleaning from the start, and move to `available` state when complete.

4.2.5 Node adoption

Overview

As part of hardware inventory lifecycle management, it is not an unreasonable need to have the capability to be able to add hardware that should be considered in-use by the Bare Metal service, that may have been deployed by another Bare Metal service installation or deployed via other means.

As such, the node adoption feature allows a user to define a node as `active` while skipping the `available` and `deploying` states, which will prevent the node from being seen by the Compute service as ready for use.

This feature is leveraged as part of the state machine workflow, where a node in `manageable` can be moved to `active` state via the `provision_state` verb `adopt`. To view the state transition capabilities, please see *Ironics State Machine*.

How it works

A node initially enrolled begins in the `enroll` state. An operator must then move the node to `manageable` state, which causes the nodes power interface to be validated. Once in `manageable` state, an operator can then explicitly choose to adopt a node.

Adoption of a node results in the validation of its `boot` interface, and upon success the process leverages what is referred to as the takeover logic. The takeover process is intended for conductors to take over the management of nodes for a conductor that has failed.

The takeover process involves the deploy interfaces `prepare` and `take_over` methods being called. These steps take specific actions such as downloading and staging the deployment kernel and ramdisk, ISO image, any required boot image, or boot ISO image and then places any PXE or virtual media configuration necessary for the node should it be required.

The adoption process makes no changes to the physical node, with the exception of operator supplied configurations where virtual media is used to boot the node under normal circumstances. An operator should ensure that any supplied configuration defining the node is sufficient for the continued operation of the node moving forward. Such as, if the node is configured to network boot via `instance_info/boot_option=netboot`, then appropriate driver specific node configuration should be set to support this capability.

Possible Risk

The main risk with this feature is that supplied configuration may ultimately be incorrect or invalid which could result in potential operational issues:

- `rebuild` verb - Rebuild is intended to allow a user to re-deploy the node to a fresh state. The risk with adoption is that the image defined when an operator adopts the node may not be the valid image for the pre-existing configuration.

If this feature is utilized for a migration from one deployment to another, and pristine original images are loaded and provided, then ultimately the risk is the same with any normal use of the `rebuild` feature, the server is effectively wiped.

- When deleting a node, the deletion or cleaning processes may fail if the incorrect deployment image is supplied in the configuration as the node may NOT have been deployed with the supplied image and driver or compatibility issues may exist as a result.

Operators will need to be cognizant of that possibility and should plan accordingly to ensure that deployment images are known to be compatible with the hardware in their environment.

- Networking - Adoption will assert no new networking configuration to the newly adopted node as that would be considered modifying the node.

Operators will need to plan accordingly and have network configuration such that the nodes will be able to network boot.

How to use

Note: The power state that the ironic-conductor observes upon the first successful power state check, as part of the transition to the `manageable` state will be enforced with a node that has been adopted. This means a node that is in `power off` state will, by default, have the power state enforced as `power off` moving forward, unless an administrator actively changes the power state using the Bare Metal service.

Requirements

Requirements for use are essentially the same as to deploy a node:

- Sufficient driver information to allow for a successful power management validation.
- Sufficient `instance_info` to pass deploy interface preparation.

Each driver may have additional requirements dependent upon the configuration that is supplied. An example of this would be defining a node to always boot from the network, which will cause the conductor to attempt to retrieve the pertinent files. Inability to do so will result in the adoption failing, and the node being placed in the `adopt failed` state.

Example

This is an example to create a new node, named `testnode`, with sufficient information to pass basic validation in order to be taken from the `manageable` state to `active` state:

```
# Explicitly set the client API version environment variable to
# 1.17, which introduces the adoption capability.
export OS_BAREMETAL_API_VERSION=1.17

baremetal node create --name testnode \
  --driver ipmi \
  --driver-info ipmi_address=<ip_address> \
  --driver-info ipmi_username=<username> \
  --driver-info ipmi_password=<password> \
  --driver-info deploy_kernel=<deploy_kernel_id_or_url> \
  --driver-info deploy_ramdisk=<deploy_ramdisk_id_or_url>

baremetal port create <node_mac_address> --node <node_uuid>

baremetal node set testnode \
  --instance-info image_source="http://localhost:8080/blankimage" \
  --instance-info capabilities="{\"boot_option\": \"local\"}"

baremetal node manage testnode --wait

baremetal node adopt testnode --wait
```

Note: In the above example, the `image_source` setting must reference a valid image or file, however that image or file can ultimately be empty.

Note: The above example utilizes a capability that defines the boot operation to be local. It is recommended to define the node as such unless network booting is desired.

Note: The above example will fail a re-deployment as a fake image is defined and no `instance_info/image_checksum` value is defined. As such any actual attempt to write the image out will fail as the `image_checksum` value is only validated at time of an actual deployment operation.

Note: A user may wish to assign an `instance_uuid` to a node, which could be used to match an instance in the Compute service. Doing so is not required for the proper operation of the Bare Metal service.

```
baremetal node set <node name or uuid> instance-uuid <uuid>
```

Note: In Newton, coupled with API version 1.20, the concept of a `network_interface` was introduced. A user of this feature may wish to add new nodes with a `network_interface` of `noop` and then change the interface at a later point and time.

Troubleshooting

Should an adoption operation fail for a node, the error that caused the failure will be logged in the nodes `last_error` field when viewing the node. This error, in the case of node adoption, will largely be due to failure of a validation step. Validation steps are dependent upon what driver is selected for the node.

Any node that is in the `adopt failed` state can have the `adopt` verb re-attempted. Example:

```
baremetal node adopt <node name or uuid>
```

If a user wishes to abort their attempt at adopting, they can then move the node back to `manageable` from `adopt failed` state by issuing the `manage` verb. Example:

```
baremetal node manage <node name or uuid>
```

If all else fails the hardware node can be removed from the Bare Metal service. The `node delete` command, which is **not** the same as setting the provision state to `deleted`, can be used while the node is in `adopt failed` state. This will delete the node without cleaning occurring to preserve the nodes current state. Example:

```
baremetal node delete <node name or uuid>
```

4.2.6 Node retirement

Overview

Retiring nodes is a natural part of a servers life cycle, for instance when the end of the warranty is reached and the physical space is needed for new deliveries to install replacement capacity.

However, depending on the type of the deployment, removing nodes from service can be a full workflow by itself as it may include steps like moving applications to other hosts, cleaning sensitive data from disks or the BMC, or tracking the dismantling of servers from their racks.

Ironic provides some means to support such workflows by allowing to tag nodes as `retired` which will prevent any further scheduling of instances, but will still allow for other operations, such as cleaning, to happen (this marks an important difference to nodes which have the `maintenance` flag set).

How to use

When it is known that a node shall be retired, set the `retired` flag on the node with:

```
baremetal node set --retired node-001
```

This can be done irrespective of the state the node is in, so in particular while the node is `active`.

Note: An exception are nodes which are in `available`. For backwards compatibility reasons, these nodes need to be moved to `manageable` first. Trying to set the `retired` flag for `available` nodes will result in an error.

Optionally, a reason can be specified when a node is retired, e.g.:

```
baremetal node set --retired node-001 \  
--retired-reason "End of warranty for delivery abc123"
```

Upon instance deletion, an `active` node with the `retired` flag set will not move to `available`, but to `manageable`. The node will hence not be eligible for scheduling of new instances.

Equally, nodes with `retired` set to `True` cannot move from `manageable` to `available`: the `provide` verb is blocked. This is to prevent accidental re-use of nodes tagged for removal from the fleet. In order to move these nodes to `available` none the less, the `retired` field needs to be removed first. This can be done via:

```
baremetal node unset --retired node-001
```

In order to facilitate the identification of nodes marked for retirement, e.g. by other teams, ironic also allows to list all nodes which have the `retired` flag set:

```
baremetal node list --retired
```


4.2.7 RAID Configuration

Overview

Ironic supports RAID configuration for bare metal nodes. It allows operators to specify the desired RAID configuration via the OpenStackClient CLI or REST API. The desired RAID configuration is applied on the bare metal during manual cleaning.

The examples described here use the OpenStackClient CLI; please see the [REST API reference](#) for their corresponding REST API requests.

Prerequisites

The bare metal node needs to use a hardware type that supports RAID configuration. RAID interfaces may implement RAID configuration either in-band or out-of-band. Software RAID is supported on all hardware, although with some caveats - see *Software RAID* for details.

In-band RAID configuration (including software RAID) is done using the Ironic Python Agent ramdisk. For in-band hardware RAID configuration, a hardware manager which supports RAID should be bundled with the ramdisk.

Whether a node supports RAID configuration could be found using the CLI command `baremetal node validate <node>`. In-band RAID is usually implemented by the agent RAID interface.

Build agent ramdisk which supports RAID configuration

For doing in-band hardware RAID configuration, Ironic needs an agent ramdisk bundled with a hardware manager which supports RAID configuration for your hardware. For example, the *DIB support for Proliant Hardware Manager* should be used for HPE Proliant Servers.

Note: For in-band software RAID, the agent ramdisk does not need to be bundled with a hardware manager as the generic hardware manager in the Ironic Python Agent already provides (basic) support for software RAID.

RAID configuration JSON format

The desired RAID configuration and current RAID configuration are represented in JSON format.

Target RAID configuration

This is the desired RAID configuration on the bare metal node. Using the OpenStackClient CLI (or REST API), the operator sets `target_raid_config` field of the node. The target RAID configuration will be applied during manual cleaning.

Target RAID configuration is a dictionary having `logical_disks` as the key. The value for the `logical_disks` is a list of JSON dictionaries. It looks like:

```
{
  "logical_disks": [
    {<desired properties of logical disk 1>},
    {<desired properties of logical disk 2>},
    ...
  ]
}
```

If the `target_raid_config` is an empty dictionary, it unsets the value of `target_raid_config` if the value was set with previous RAID configuration done on the node.

Each dictionary of logical disk contains the desired properties of logical disk supported by the hardware type. These properties are discoverable by:

```
baremetal driver raid property list <driver name>
```

Mandatory properties

These properties must be specified for each logical disk and have no default values:

- `size_gb` - Size (Integer) of the logical disk to be created in GiB. `MAX` may be specified if the logical disk should use all of the remaining space available. This can be used only when backing physical disks are specified (see below).
- `raid_level` - RAID level for the logical disk. Ironic supports the following RAID levels: 0, 1, 2, 5, 6, 1+0, 5+0, 6+0.

Optional properties

These properties have default values and they may be overridden in the specification of any logical disk. None of these options are supported for software RAID.

- `volume_name` - Name of the volume. Should be unique within the Node. If not specified, volume name will be auto-generated.
- `is_root_volume` - Set to `true` if this is the root volume. At most one logical disk can have this set to `true`; the other logical disks must have this set to `false`. The `root device hint` will be saved, if the RAID interface is capable of retrieving it. This is `false` by default.

Backing physical disk hints

These hints are specified for each logical disk to let Ironic find the desired disks for RAID configuration. This is machine-independent information. This serves the use-case where the operator doesn't want to provide individual details for each bare metal node. None of these options are supported for software RAID.

- `share_physical_disks` - Set to `true` if this logical disk can share physical disks with other logical disks. The default value is `false`, except for software RAID which always shares disks.
- `disk_type` - `hdd` or `ssd`. If this is not specified, disk type will not be a criterion to find backing physical disks.

- `interface_type` - sata or scsi or sas. If this is not specified, interface type will not be a criterion to find backing physical disks.
- `number_of_physical_disks` - Integer, number of disks to use for the logical disk. Defaults to minimum number of disks required for the particular RAID level, except for software RAID which always spans all disks.

Backing physical disks

These are the actual machine-dependent information. This is suitable for environments where the operator wants to automate the selection of physical disks with a 3rd-party tool based on a wider range of attributes (eg. S.M.A.R.T. status, physical location). The values for these properties are hardware dependent.

- `controller` - The name of the controller as read by the RAID interface. In order to trigger the setup of a Software RAID via the Ironic Python Agent, the value of this property needs to be set to `software`.
- `physical_disks` - A list of physical disks to use as read by the RAID interface.

For software RAID `physical_disks` is a list of device hints in the same format as used for *Specifying the disk for deployment (root device hints)*. The number of provided hints must match the expected number of backing devices (repeat the same hint if necessary).

Note: If properties from both Backing physical disk hints or Backing physical disks are specified, they should be consistent with each other. If they are not consistent, then the RAID configuration will fail (because the appropriate backing physical disks could not be found).

Examples for `target_raid_config`

Example 1. Single RAID disk of RAID level 5 with all of the space available. Make this the root volume to which Ironic deploys the image:

```
{
  "logical_disks": [
    {
      "size_gb": "MAX",
      "raid_level": "5",
      "is_root_volume": true
    }
  ]
}
```

Example 2. Two RAID disks. One with RAID level 5 of 100 GiB and make it root volume and use SSD. Another with RAID level 1 of 500 GiB and use HDD:

```
{
  "logical_disks": [
    {
      "size_gb": 100,
```

(continues on next page)

(continued from previous page)

```

    "raid_level": "5",
    "is_root_volume": true,
    "disk_type": "ssd"
  },
  {
    "size_gb": 500,
    "raid_level": "1",
    "disk_type": "hdd"
  }
]
}

```

Example 3. Single RAID disk. I know which disks and controller to use:

```

{
  "logical_disks": [
    {
      "size_gb": 100,
      "raid_level": "5",
      "controller": "Smart Array P822 in Slot 3",
      "physical_disks": ["6I:1:5", "6I:1:6", "6I:1:7"],
      "is_root_volume": true
    }
  ]
}

```

Example 4. Using backing physical disks:

```

{
  "logical_disks": [
    {
      "size_gb": 50,
      "raid_level": "1+0",
      "controller": "RAID.Integrated.1-1",
      "volume_name": "root_volume",
      "is_root_volume": true,
      "physical_disks": [
        "Disk.Bay.0:Encl.Int.0-1:RAID.Integrated.1-1",
        "Disk.Bay.1:Encl.Int.0-1:RAID.Integrated.1-1"
      ]
    },
    {
      "size_gb": 100,
      "raid_level": "5",
      "controller": "RAID.Integrated.1-1",
      "volume_name": "data_volume",
      "physical_disks": [
        "Disk.Bay.2:Encl.Int.0-1:RAID.Integrated.1-1",
        "Disk.Bay.3:Encl.Int.0-1:RAID.Integrated.1-1",
        "Disk.Bay.4:Encl.Int.0-1:RAID.Integrated.1-1"
      ]
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    ]
  }
]
}

```

Example 5. Software RAID with two RAID devices:

```

{
  "logical_disks": [
    {
      "size_gb": 100,
      "raid_level": "1",
      "controller": "software"
    },
    {
      "size_gb": "MAX",
      "raid_level": "0",
      "controller": "software"
    }
  ]
}

```

Example 6. Software RAID, limiting backing block devices to exactly two devices with the size exceeding 100 GiB:

```

{
  "logical_disks": [
    {
      "size_gb": "MAX",
      "raid_level": "0",
      "controller": "software",
      "physical_disks": [
        {"size": "> 100"},
        {"size": "> 100"}
      ]
    }
  ]
}

```

Current RAID configuration

After target RAID configuration is applied on the bare metal node, Ironic populates the current RAID configuration. This is populated in the `raid_config` field in the Ironic node. This contains the details about every logical disk after they were created on the bare metal node. It contains details like RAID controller used, the backing physical disks used, WWN of each logical disk, etc. It also contains information about each physical disk found on the bare metal node.

To get the current RAID configuration:

```
baremetal node show <node-uuid-or-name>
```

Workflow

- Operator configures the bare metal node with a hardware type that has a RAIDInterface other than no-raid. For instance, for Software RAID, this would be agent.
- For in-band RAID configuration, operator builds an agent ramdisk which supports RAID configuration by bundling the hardware manager with the ramdisk. See *Build agent ramdisk which supports RAID configuration* for more information.
- Operator prepares the desired target RAID configuration as mentioned in *Target RAID configuration*. The target RAID configuration is set on the Ironic node:

```
baremetal node set <node-uuid-or-name> \  
  --target-raid-config <JSON file containing target RAID configuration>
```

The CLI command can accept the input from standard input also:

```
baremetal node set <node-uuid-or-name> \  
  --target-raid-config -
```

- Create a JSON file with the RAID clean steps for manual cleaning. Add other clean steps as desired:

```
[{  
  "interface": "raid",  
  "step": "delete_configuration"  
},  
{  
  "interface": "raid",  
  "step": "create_configuration"  
}]
```

Note: create_configuration doesnt remove existing disks. It is recommended to add delete_configuration before create_configuration to make sure that only the desired logical disks exist in the system after manual cleaning.

- Bring the node to manageable state and do a clean action to start cleaning on the node:

```
baremetal node clean <node-uuid-or-name> \  
  --clean-steps <JSON file containing clean steps created above>
```

- After manual cleaning is complete, the current RAID configuration is reported in the raid_config field when running:

```
baremetal node show <node-uuid-or-name>
```

Software RAID

Building Linux software RAID in-band (via the Ironic Python Agent ramdisk) is supported starting with the Train release. It is requested by using the agent RAID interface and RAID configuration with all controllers set to software. You can find a software RAID configuration example in *Examples for target_raid_config*.

There are certain limitations to be aware of:

- Only the mandatory properties (plus the required controller property) from *Target RAID configuration* are currently supported.
- The number of created Software RAID devices must be 1 or 2. If there is only one Software RAID device, it has to be a RAID-1. If there are two, the first one has to be a RAID-1, while the RAID level for the second one can be 0, 1, 1+0, 5, or 6. As the first RAID device will be the deployment device, enforcing a RAID-1 reduces the risk of ending up with a non-booting node in case of a disk failure.
- Building RAID will fail if the target disks are already partitioned. Wipe the disks using e.g. the `erase_devices_metadata` clean step before building RAID:

```
[{
  "interface": "raid",
  "step": "delete_configuration"
},
{
  "interface": "deploy",
  "step": "erase_devices_metadata"
},
{
  "interface": "raid",
  "step": "create_configuration"
}]
```

- If local boot is going to be used, the final instance image must have the `mdadm` utility installed and needs to be able to detect software RAID devices at boot time (which is usually done by having the RAID drivers embedded in the images `initrd`).
- Regular cleaning will not remove RAID configuration (similarly to hardware RAID). To destroy RAID run the `delete_configuration` manual clean step.
- There is no support for partition images, only whole-disk images are supported with Software RAID. See *Add images to the Image service*.
- In UEFI mode, the Ironic Python Agent creates EFI system partitions (ESPs) for the bootloader and the boot configuration (`grub.cfg` or `grubenv`) on all holder devices. The content of these partitions is populated upon deployment from the deployed user image. Depending on how the partitions are mounted, the content of the partitions may get out of sync, e.g. when new kernels are installed or the bootloader is updated, so measures to keep these partitions in sync need to be taken.

Image requirements

Since Ironic needs to perform additional steps when deploying nodes with software RAID, there are some requirements the deployed images need to fulfill. Up to and including the Train release, the image needs to have its root file system on the first partition. Starting with Ussuri, the image can also have additional metadata to point Ironic to the partition with the root file system: for this, the image needs to set the `rootfs_uuid` property with the file system UUID of the root file system. One way to extract this UUID from an existing image is to download the image, mount it as a loopback device, and use `blkid`:

```
$ sudo losetup -f
$ sudo losetup /dev/loop0 /tmp/myimage.raw
$ sudo kpartx -a /dev/loop0
$ blkid
```

The pre-Ussuri approach, i.e. to have the root file system on the first partition, is kept as a fallback and hence allows software RAID deployments where Ironic does not have access to any image metadata (e.g. Ironic stand-alone).

Using RAID in nova flavor for scheduling

The operator can specify the `raid_level` capability in nova flavor for node to be selected for scheduling:

```
openstack flavor set my-baremetal-flavor --property capabilities:raid_level=
↪ "1+0"
```

Developer documentation

In-band RAID configuration is done using IPA ramdisk. IPA ramdisk has support for pluggable hardware managers which can be used to extend the functionality offered by IPA ramdisk using stevedore plugins. For more information, see Ironic Python Agent [Hardware Manager](#) documentation.

The hardware manager that supports RAID configuration should do the following:

1. Implement a method named `create_configuration`. This method creates the RAID configuration as given in `target_raid_config`. After successful RAID configuration, it returns the current RAID configuration information which ironic uses to set `node.raid_config`.
2. Implement a method named `delete_configuration`. This method deletes all the RAID disks on the bare metal.
3. Return these two clean steps in `get_clean_steps` method with priority as 0. Example:

```
return [{ 'step': 'create_configuration',
          'interface': 'raid',
          'priority': 0 },
        { 'step': 'delete_configuration',
          'interface': 'raid',
          'priority': 0 }]
```


4.2.8 BIOS Configuration

Overview

The Bare Metal service supports BIOS configuration for bare metal nodes. It allows administrators to retrieve and apply the desired BIOS settings via CLI or REST API. The desired BIOS settings are applied during manual cleaning.

Prerequisites

Bare metal servers must be configured by the administrator to be managed via ironic hardware type that supports BIOS configuration.

Enabling hardware types

Enable a specific hardware type that supports BIOS configuration. Refer to *Enabling drivers and hardware types* for how to enable a hardware type.

Enabling hardware interface

To enable the bios interface:

```
[DEFAULT]
enabled_bios_interfaces = no-bios
```

Append the actual bios interface name supported by the enabled hardware type to `enabled_bios_interfaces` with comma separated values in `ironic.conf`.

All available in-tree bios interfaces are listed in `setup.cfg` file in the source code tree, for example:

```
ironic.hardware.interfaces.bios =
    fake = ironic.drivers.modules.fake:FakeBIOS
    no-bios = ironic.drivers.modules.noop:NoBIOS
```

Retrieve BIOS settings

To retrieve the cached BIOS configuration from a specified node:

```
$ baremetal node bios setting list <node>
```

BIOS settings are cached on each node cleaning operation or when settings have been applied successfully via BIOS cleaning steps. The return of above command is a table of last cached BIOS settings from specified node. If `-f json` is added as suffix to above command, it returns BIOS settings as following:

```
[
  {
    "setting name":
    {
```

(continues on next page)

(continued from previous page)

```
        "name": "setting name",
        "value": "value"
    }
},
{
    "setting name":
    {
        "name": "setting name",
        "value": "value"
    }
},
...
]
```

To get a specified BIOS setting for a node:

```
$ baremetal node bios setting show <node> <setting-name>
```

If `-f json` is added as suffix to above command, it returns BIOS settings as following:

```
{
  "setting name":
  {
    "name": "setting name",
    "value": "value"
  }
}
```

Configure BIOS settings

Two *Manual cleaning* steps are available for managing nodes BIOS settings:

Factory reset

This cleaning step resets all BIOS settings to factory default for a given node:

```
{
  "target": "clean",
  "clean_steps": [
    {
      "interface": "bios",
      "step": "factory_reset"
    }
  ]
}
```

The `factory_reset` cleaning step does not require any arguments, as it resets all BIOS settings to factory defaults.

Apply BIOS configuration

This cleaning step applies a set of BIOS settings for a node:

```
{
  "target": "clean",
  "clean_steps": [
    {
      "interface": "bios",
      "step": "apply_configuration",
      "args": {
        "settings": [
          {
            "name": "name",
            "value": "value"
          },
          {
            "name": "name",
            "value": "value"
          }
        ]
      }
    }
  ]
}
```

The representation of `apply_configuration` cleaning step follows the same format of *Manual cleaning*. The desired BIOS settings can be provided via the `settings` argument which contains a list of BIOS options to be applied, each BIOS option is a dictionary with `name` and `value` keys.

To check whether the desired BIOS configuration is set properly, use the command mentioned in the *Retrieve BIOS settings* section.

Note: When applying BIOS settings to a node, vendor-specific driver may take the given BIOS settings from the argument and compare them with the current BIOS settings on the node and only apply when there is a difference.

4.2.9 Rescue Mode

Overview

The Bare Metal Service supports putting nodes in rescue mode using hardware types that support rescue interfaces. The hardware types utilizing ironic-python-agent with PXE/Virtual Media based boot interface can support rescue operation when configured appropriately.

Note: The rescue operation is currently supported only when tenant networks use DHCP to obtain IP addresses.

Rescue operation can be used to boot nodes into a rescue ramdisk so that the rescue user can access the node, in order to provide the ability to access the node in case access to OS is not possible. For example, if there is a need to perform manual password reset or data recovery in the event of some failure, rescue operation can be used.

Configuring The Bare Metal Service

Configure the Bare Metal Service appropriately so that the service has the information needed to boot the ramdisk before a user tries to initiate rescue operation. This will differ somewhat between different deploy environments, but an example of how to do this is outlined below:

1. Create and configure ramdisk that supports rescue operation. Please see *Building or downloading a deploy ramdisk image* for detailed instructions to build a ramdisk.
2. Configure a network to use for booting nodes into the rescue ramdisk in neutron, and note the UUID or name of this network. This is required if you're using the neutron DHCP provider and have Bare Metal Service managing ramdisk booting (the default). This can be the same network as your cleaning or tenant network (for flat network). For an example of how to configure new networks with Bare Metal Service, see the *Configure the Networking service for bare metal provisioning* documentation.
3. Add the unique name or UUID of your rescue network to `ironic.conf`:

```
[neutron]
rescuing_network=<RESCUE_UUID_OR_NAME>
```

Note: This can be set per node via `driver_info[rescuing_network]`

4. Restart the ironic conductor service.
5. Specify a rescue kernel and ramdisk or rescue ISO compatible with the nodes driver for pxe based boot interface or virtual-media based boot interface respectively.

Example for pxe based boot interface:

```
baremetal node set $NODE_UUID \
  --driver-info rescue_ramdisk=$RESCUE_INITRD_UUID \
  --driver-info rescue_kernel=$RESCUE_VMLINUZ_UUID
```

See *Add images to the Image service* for details. If you are not using Image service, it is possible to provide images to Bare Metal service via hrefs.

After this, The Bare Metal Service should be ready for rescue operation. Test it out by attempting to rescue an active node and connect to the instance using ssh, as given below:

```
baremetal node rescue $NODE_UUID \
  --rescue-password <PASSWORD> --wait

ssh rescue@$INSTANCE_IP_ADDRESS
```

To move a node back to active state after using rescue mode you can use `unrescue`. Please unmount any filesystems that were manually mounted before proceeding with `unrescue`. The node `unrescue` can be done as given below:

```
baremetal node unrescue $NODE_UUID
```

rescue and unrescue operations can also be triggered via the Compute Service using the following commands:

```
openstack server rescue --password <password> <server>
openstack server unrescue <server>
```

4.2.10 Boot From Volume

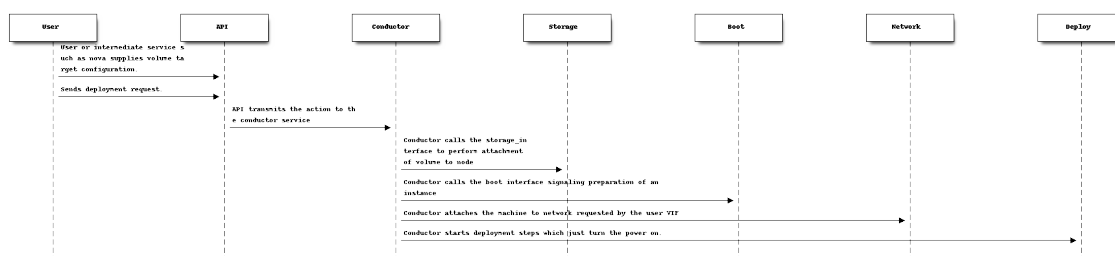
Overview

The Bare Metal service supports booting from a Cinder iSCSI volume as of the Pike release. This guide will primarily deal with this use case, but will be updated as more paths for booting from a volume, such as FCoE, are introduced.

The boot from volume is supported on both legacy BIOS and UEFI (iPXE binary for EFI booting) boot mode. We need to perform with suitable images which will be created by diskimage-builder tool.

How this works - From Ironics point of view

In essence, ironic sets the stage for the process, by providing the required information to the boot interface to facilitate the configuration of the the node OR the iPXE boot templates such that the node CAN be booted.



In this example, the boot interface does the heavy lifting. For drivers the `irmc` and `ilo` hardware types with hardware type specific boot interfaces, they are able to signal via an out of band mechanism to the baremetal nodes BMC that the integrated iSCSI initiators are to connect to the supplied volume target information.

In most hardware this would be the network cards of the machine.

In the case of the `ipxe` boot interface, templates are created on disk which point to the `iscsi` target information that was either submitted as part of the volume target, or when integrated with Nova, what was requested as the baremetals boot from volume disk upon requesting the instance.

In terms of network access, both interface methods require connectivity to the `iscsi` target. In the vendor driver specific path, additional network configuration options may be available to allow separation of standard network traffic and instance network traffic. In the `iPXE` case, this is not possible as the OS userspace re-configures the `iSCSI` connection after detection inside the OS ramdisk boot.

An `iPXE` user *may* be able to leverage multiple VIFs, one specifically set to be set with `pxe_enabled` to handle the initial instance boot and back-end storage traffic where as external facing network traffic

occurs on a different interface. This is a common pattern in iSCSI based deployments in the physical realm.

Prerequisites

Currently booting from a volume requires:

- Bare Metal service version 9.0.0
- Bare Metal API microversion 1.33 or later
- A driver that utilizes the *PXE boot mechanism*. Currently booting from a volume is supported by the reference drivers that utilize PXE boot mechanisms when iPXE is enabled.
- iPXE is an explicit requirement, as it provides the mechanism that attaches and initiates booting from an iSCSI volume.
- Metadata services need to be configured and available for the instance images to obtain configuration such as keys. Configuration drives are not supported due to minimum disk extension sizes.

Conductor Configuration

In `ironic.conf`, you can specify a list of enabled storage interfaces. Check `[DEFAULT]enabled_storage_interfaces` in your `ironic.conf` to ensure that your desired interface is enabled. For example, to enable the `cinder` and `noop` storage interfaces:

```
[DEFAULT]
enabled_storage_interfaces = cinder,noop
```

If you want to specify a default storage interface rather than setting the storage interface on a per node basis, set `[DEFAULT]default_storage_interface` in `ironic.conf`. The `default_storage_interface` will be used for any node that doesn't have a storage interface defined.

Node Configuration

Storage Interface

You will need to specify what storage interface the node will use to handle storage operations. For example, to set the storage interface to `cinder` on an existing node:

```
baremetal node set --storage-interface cinder $NODE_UUID
```

A default storage interface can be specified in `ironic.conf`. See the *Conductor Configuration* section for details.

iSCSI Configuration

In order for a bare metal node to boot from an iSCSI volume, the `iscsi_boot` capability for the node must be set to `True`. For example, if you want to update an existing node to boot from volume:

```
baremetal node set --property capabilities=iscsi_boot:True $NODE_UUID
```

You will also need to create a volume connector for the node, so the storage interface will know how to communicate with the node for storage operation. In the case of iSCSI, you will need to provide an iSCSI Qualifying Name (IQN) that is unique to your SAN. For example, to create a volume connector for iSCSI:

```
baremetal volume connector create \  
    --node $NODE_UUID --type iqn --connector-id iqn.2017-08.org.  
    ↪openstack.$NODE_UUID
```

Image Creation

We use `disk-image-create` in `diskimage-builder` tool to create images for boot from volume feature. Some required elements for this mechanism for corresponding boot modes are as following:

- Legacy BIOS boot mode: `iscsi-boot` element.
- UEFI boot mode: `iscsi-boot` and `block-device-efi` elements.

An example below:

```
export IMAGE_NAME=<image_name>  
export DIB_CLOUD_INIT_DATASOURCES="ConfigDrive, OpenStack"  
disk-image-create centos7 vm cloud-init-datasources dhcp-all-interfaces iscsi-  
    ↪boot dracut-regenerate block-device-efi -o $IMAGE_NAME
```

Note:

- For CentOS images, we must add dependent element named `dracut-regenerate` during image creation. Otherwise, the image creation will fail with an error.
- For Ubuntu images, we only support `iscsi-boot` element without `dracut-regenerate` element during image creation.

Advanced Topics

Use without the Compute Service

As discussed in other sections, the Bare Metal service has a concept of a *connector* that is used to represent an interface that is intended to be utilized to attach the remote volume.

In addition to the connectors, we have a concept of a *target* that can be defined via the API. While a user of this feature through the Compute service would automatically have a new target record created for them, it is not explicitly required, and can be performed manually.

A target record can be created using a command similar to the example below:

```
baremetal volume target create \  
    --node $NODE_UUID --type iscsi --boot-index 0 --volume $VOLUME_UUID
```

Note: A `boot-index` value of `0` represents the boot volume for a node. As the `boot-index` is per-node in sequential order, only one boot volume is permitted for each node.

Use Without Cinder

In the Rocky release, an `external` storage interface is available that can be utilized without a Block Storage Service installation.

Under normal circumstances the `cinder` storage interface interacts with the Block Storage Service to orchestrate and manage attachment and detachment of volumes from the underlying block service system.

The `external` storage interface contains the logic to allow the Bare Metal service to determine if the Bare Metal node has been requested with a remote storage volume for booting. This is in contrast to the default `noop` storage interface which does not contain logic to determine if the node should or could boot from a remote volume.

It must be noted that minimal configuration or value validation occurs with the `external` storage interface. The `cinder` storage interface contains more extensive validation, that is likely un-necessary in a `external` scenario.

Setting the external storage interface:

```
baremetal node set --storage-interface external $NODE_UUID
```

Setting a volume:

```
baremetal volume target create --node $NODE_UUID \  
    --type iscsi --boot-index 0 --volume-id $VOLUME_UUID \  
    --property target_iqn="iqn.2010-10.com.example:vol-X" \  
    --property target_lun="0" \  
    --property target_portal="192.168.0.123:3260" \  
    --property auth_method="CHAP" \  
    --property auth_username="ABC" \  
    --property auth_password="XYZ" \  
    \
```

Ensure that no `image_source` is defined:

```
baremetal node unset \  
    --instance-info image_source $NODE_UUID
```

Deploy the node:

```
baremetal node deploy $NODE_UUID
```

Upon deploy, the boot interface for the baremetal node will attempt to either create iPXE configuration OR set boot parameters out-of-band via the management controller. Such action is boot interface specific and may not support all forms of volume target configuration. As of the Rocky release, the bare metal

service does not support writing an Operating System image to a remote boot from volume target, so that also must be ensured by the user in advance.

Records of volume targets are removed upon the node being undeployed, and as such are not persistent across deployments.

Cinder Multi-attach

Volume multi-attach is a function that is commonly performed in computing clusters where dedicated storage subsystems are utilized. For some time now, the Block Storage service has supported the concept of multi-attach. However, the Compute service, as of the Pike release, does not yet have support to leverage multi-attach. Concurrently, multi-attach requires the backend volume driver running as part of the Block Storage service to contain support for multi-attach volumes.

When support for storage interfaces was added to the Bare Metal service, specifically for the `cinder` storage interface, the concept of volume multi-attach was accounted for, however has not been fully tested, and is unlikely to be fully tested until there is Compute service integration as well as volume driver support.

The data model for storage of volume targets in the Bare Metal service has no constraints on the same target volume from being utilized. When interacting with the Block Storage service, the Bare Metal service will prevent the use of volumes that are being reported as `in-use` if they do not explicitly support multi-attach.

4.2.11 Multi-tenancy in the Bare Metal service

Overview

It is possible to use dedicated tenant networks for provisioned nodes, which extends the current Bare Metal service capabilities of providing flat networks. This works in conjunction with the Networking service to allow provisioning of nodes in a separate provisioning network. The result of this is that multiple tenants can use nodes in an isolated fashion. However, this configuration does not support trunk ports belonging to multiple networks.

Concepts

Network interfaces

Network interface is one of the driver interfaces that manages network switching for nodes. There are 3 network interfaces available in the Bare Metal service:

- `noop` interface is used for standalone deployments, and does not perform any network switching;
- `flat` interface places all nodes into a single provider network that is pre-configured on the Networking service and physical equipment. Nodes remain physically connected to this network during their entire life cycle.
- `neutron` interface provides tenant-defined networking through the Networking service, separating tenant networks from each other and from the provisioning and cleaning provider networks. Nodes will move between these networks during their life cycle. This interface requires Networking service support for the switches attached to the baremetal servers so they can be programmed.

Local link connection

The Bare Metal service allows `local_link_connection` information to be associated with Bare Metal ports. This information is provided to the Networking services ML2 driver when a Virtual Interface (VIF) is attached. The ML2 driver uses the information to plug the specified port to the tenant network.

Table 21: `local_link_connection` fields

Field	Description
<code>switch_id</code>	Required. Identifies a switch and can be a MAC address or an OpenFlow-based <code>datapath_id</code> .
<code>port_id</code>	Required. Port ID on the switch/Smart NIC, for example, Gig0/1, rep0-0.
<code>switch_info</code>	Optional. Used to distinguish different switch models or other vendor-specific identifier. Some ML2 plugins may require this field.
<code>hostname</code>	Required in case of a Smart NIC port. Hostname of Smart NIC device.

Note: This isn't applicable to Infiniband ports because the network topology is discoverable by the Infiniband Subnet Manager. If specified, `local_link_connection` information will be ignored. If port is Smart NIC port then:

1. `port_id` is the representor port name on the Smart NIC.
 2. `switch_id` is not mandatory.
-

Physical networks

A Bare Metal port may be associated with a physical network using its `physical_network` field. The Bare Metal service uses this information when mapping between virtual ports in the Networking service and physical ports and port groups in the Bare Metal service. A port's `physical_network` field is optional, and if not set then any virtual port may be mapped to that port, provided that no free Bare Metal port with a suitable physical network assignment exists.

The physical network of a port group is defined by the physical network of its constituent ports. The Bare Metal service ensures that all ports in a port group have the same value in their `physical_network` field.

When attaching a virtual interface (VIF) to a node, the following ordered criteria are used to select a suitable unattached port or port group:

- Require ports or port groups to not have a physical network or to have a physical network that matches one of the VIFs allowed physical networks.
- Prefer ports and port groups that have a physical network to ports and port groups that do not have a physical network.
- Prefer port groups to ports. Prefer ports with PXE enabled.

Configuring the Bare Metal service

See the *Configure tenant networks* section in the installation guide for the Bare Metal service.

Configuring nodes

1. Ensure that your python-ironicclient version and requested API version are sufficient for your requirements.
 - Multi-tenancy support was added in API version 1.20, and is supported by python-ironicclient version 1.5.0 or higher.
 - Physical network support for ironic ports was added in API version 1.34, and is supported by python-ironicclient version 1.15.0 or higher.
 - Smart NIC support for ironic ports was added in API version 1.53, and is supported by python-ironicclient version 2.7.0 or higher.

The following examples assume you are using python-ironicclient version 2.7.0 or higher.

Export the following variable:

```
export OS_BAREMETAL_API_VERSION=<API version>
```

2. The nodes `network_interface` field should be set to a valid network interface. Valid interfaces are listed in the `[DEFAULT]/enabled_network_interfaces` configuration option in the ironic-conductors configuration file. Set it to `neutron` to use the Networking services ML2 driver:

```
baremetal node create --network-interface neutron --driver ipmi
```

Note: If the `[DEFAULT]/default_network_interface` configuration option is set, the `--network-interface` option does not need to be specified when creating the node.

3. To update an existing nodes network interface to `neutron`, use the following commands:

```
baremetal node set $NODE_UUID_OR_NAME \
  --network-interface neutron
```

4. Create a port as follows:

```
baremetal port create $HW_MAC_ADDRESS --node $NODE_UUID \
  --local-link-connection switch_id=$SWITCH_MAC_ADDRESS \
  --local-link-connection switch_info=$SWITCH_HOSTNAME \
  --local-link-connection port_id=$SWITCH_PORT \
  --pxe-enabled true \
  --physical-network physnet1
```

An Infiniband port requires client ID, while local link connection information will be populated by Infiniband Subnet Manager. The client ID consists of <12-byte vendor prefix>:<8 byte port GUID>. There is no standard process for deriving the ports MAC address (`$HW_MAC_ADDRESS`); it is vendor specific. For example, Mellanox ConnectX Family Devices prefix is `ff:00:00:00:00:02:00:00:02:c9:00`. If port GUID was `f4:52:14:03:00:38:39:81` the client ID would be `ff:00:00:00:00:02:00:00:02:c9:00:f4:52:14:03:00:38:39:81`. Mellanox

ConnectX Family Devices `HW_MAC_ADDRESS` consists of 6 bytes; the port GUIDs lower 3 and higher 3 bytes. In this example it would be `f4:52:14:38:39:81`. Putting it all together, create an Infiniband port as follows:

```
baremetal port create $HW_MAC_ADDRESS --node $NODE_UUID \  
  --pxe-enabled true \  
  --extra client-id=$CLIENT_ID \  
  --physical-network physnet1
```

5. Create a Smart NIC port as follows:

```
baremetal port create $HW_MAC_ADDRESS --node $NODE_UUID \  
  --local-link-connection hostname=$HOSTNAME \  
  --local-link-connection port_id=$REP_NAME \  
  --pxe-enabled true \  
  --physical-network physnet1 \  
  --is-smartnic
```

A Smart NIC port requires `hostname` which is the hostname of the Smart NIC, and `port_id` which is the representor port name within the Smart NIC.

6. Check the port configuration:

```
baremetal port show $PORT_UUID
```

After these steps, the provisioning of the created node will happen in the provisioning network, and then the node will be moved to the tenant network that was requested.

Configuring the Networking service

In addition to configuring the Bare Metal service some additional configuration of the Networking service is required to ensure ports for bare metal servers are correctly programmed. This configuration will be determined by the Bare Metal service network interfaces you have enabled and which top of rack switches you have in your environment.

flat network interface

In order for Networking service ports to correctly operate with the Bare Metal service `flat` network interface the `baremetal` ML2 mechanism driver from `networking-baremetal` needs to be loaded into the Networking service configuration. This driver understands that the switch should be already configured by the admin, and will mark the networking service ports as successfully bound as nothing else needs to be done.

1. Install the `networking-baremetal` library

```
$ pip install networking-baremetal
```

2. Enable the `baremetal` driver in the Networking service ML2 configuration file

```
[ml2]  
mechanism_drivers = ovs,baremetal
```

neutron network interface

The neutron network interface allows the Networking service to program the physical top of rack switches for the bare metal servers. To do this an ML2 mechanism driver which supports the `baremetal_VNIC` type for the make and model of top of rack switch in the environment must be installed and enabled.

This is a list of known top of rack ML2 mechanism drivers which work with the neutron network interface:

Cisco Nexus 9000 series To install and configure this ML2 mechanism driver see [Nexus Mechanism Driver Installation Guide](#).

FUJITSU CFX2000 `networking-fujitsu` ML2 driver supports this switch. The documentation is available [here](#).

Networking Generic Switch This is an ML2 mechanism driver built for testing against virtual bare metal environments and some switches that are not covered by hardware specific ML2 mechanism drivers. More information is available in the projects [README](#).

4.2.12 Port groups support

The Bare Metal service supports static configuration of port groups (bonds) in the instances via `config-drive`. See [kernel documentation on bonding](#) to see why it may be useful and how it is setup in linux. The sections below describe how to make use of them in the Bare Metal service.

Switch-side configuration

If port groups are desired in the ironic deployment, they need to be configured on the switches. It needs to be done manually, and the mode and properties configured on the switch have to correspond to the mode and properties that will be configured on the ironic side, as bonding mode and properties may be named differently on your switch, or have possible values different from the ones described in [kernel documentation on bonding](#). Please refer to your switch configuration documentation for more details.

Provisioning and cleaning cannot make use of port groups if they need to boot the deployment ramdisk via (i)PXE. If your switches or desired port group configuration do not support port group fallback, which will allow port group members to be used by themselves, you need to set `port_groups_standalone_ports_supported` value to be `False` in ironic, as it is `True` by default.

Physical networks

If any port in a port group has a physical network, then all ports in that port group must have the same physical network.

In order to change the physical network of the ports in a port group, all ports must first be removed from the port group, before changing their physical networks (to the same value), then adding them back to the port group.

See [physical networks](#) for further information on using physical networks in the Bare Metal service.

Port groups configuration in the Bare Metal service

Port group configuration is supported in ironic API microversions 1.26, the CLI commands below specify it for completeness.

1. When creating a port group, the node to which it belongs must be specified, along with, optionally, its name, address, mode, properties, and if it supports fallback to standalone ports:

```
baremetal port group create \  
--node $NODE_UUID --name test --address fa:ab:25:48:fd:ba --mode 802.3ad \  
--property miimon=100 --property xmit_hash_policy="layer2+3" \  
--support-standalone-ports
```

A port group can also be updated with `baremetal port group set` command, see its help for more details.

If an address is not specified, the port group address on the deployed instance will be the same as the address of the neutron port that is attached to the port group. If the neutron port is not attached, the port group will not be configured.

Note: In standalone mode, port groups have to be configured manually. It can be done either statically inside the image, or by generating the configdrive and adding it to the nodes `instance_info`. For more information on how to configure bonding via configdrive, refer to [cloud-init documentation](#) and [code](#). cloud-init version 0.7.7 or later is required for bonding configuration to work.

The following is a simple sample for configuring bonding via configdrive:

When booting an instance, it needs to add user-data file for configuring bonding via `--user-data` option. For example:

```
{  
  "networks": [  
    {  
      "type": "physical",  
      "name": "eth0",  
      "mac_address": "fa:ab:25:48:fd:ba"  
    },  
    {  
      "type": "physical",  
      "name": "eth1",  
      "mac_address": "fa:ab:25:48:fd:ab"  
    },  
    {  
      "type": "bond",  
      "name": "bond0",  
      "bond_interfaces": [  
        "eth0", "eth1"  
      ],  
      "mode": "active-backup"  
    }  
  ]  
}
```

If the port groups address is not explicitly set in standalone mode, it will be set automatically by the process described in [kernel documentation on bonding](#).

During interface attachment, port groups have higher priority than ports, so they will be used first. (It is not yet possible to specify which one is desired, a port group or a port, in an interface attachment request). Port groups that don't have any ports will be ignored.

The mode and properties values are described in the [kernel documentation on bonding](#). The default port group mode is `active-backup`, and this default can be changed by setting the `[DEFAULT]default_portgroup_mode` configuration option in the ironic API service configuration file.

2. Associate ports with the created port group.

It can be done on port creation:

```
baremetal port create \  
--node $NODE_UUID --address fa:ab:25:48:fd:ba --port-group test
```

Or by updating an existing port:

```
baremetal port set $PORT_UUID --port-group $PORT_GROUP_UUID
```

When updating a port, the node associated with the port has to be in `enroll`, `manageable`, or `inspecting` states. A port group can have the same or different address as individual ports.

3. Boot an instance (or node directly, in case of using standalone ironic) providing an image that has cloud-init version 0.7.7 or later and supports bonding.

When the deployment is done, you can check that the port group is set up properly by running the following command in the instance:

```
cat /proc/net/bonding/bondX
```

where `X` is a number autogenerated by cloud-init for each configured port group, in no particular order. It starts with 0 and increments by 1 for every configured port group.

Link aggregation/teaming on windows

Portgroups are supported for Windows Server images, which can be created by [Building images for Windows](#) instruction.

You can customise an instance after it is launched along with [script file](#) in [Configuration of Instance](#) and selected `Configuration Drive` option. Then ironic virt driver will generate network metadata and add all the additional information, such as bond mode, transmit hash policy, MII link monitoring interval, and of which links the bond consists. The information in `InstanceMetadata` will be used afterwards to generate the config drive.

4.2.13 Configuring Web or Serial Console

Overview

There are two types of console which are available in Bare Metal service, one is web console (*Node web console*) which is available directly from web browser, another is serial console (*Node serial console*).

Node web console

The web console can be configured in Bare Metal service in the following way:

- Install shellinabox in ironic conductor node. For RHEL/CentOS, shellinabox package is not present in base repositories, user must enable EPEL repository, you can find more from [FedoraProject page](#).

Note: shellinabox is no longer maintained by the authorized author. [This](#) is a fork of the project on GitHub that aims to continue with maintenance of the shellinabox project.

Installation example:

Ubuntu:

```
sudo apt-get install shellinabox
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install shellinabox
```

You can find more about shellinabox on the [shellinabox page](#).

You can optionally use the SSL certificate in shellinabox. If you want to use the SSL certificate in shellinabox, you should install openssl and generate the SSL certificate.

1. Install openssl, for example:

Ubuntu:

```
sudo apt-get install openssl
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install openssl
```

2. Generate the SSL certificate, here is an example, you can find more about openssl on the [openssl page](#):

```
cd /tmp/ca
openssl genrsa -des3 -out my.key 1024
openssl req -new -key my.key -out my.csr
cp my.key my.key.org
openssl rsa -in my.key.org -out my.key
openssl x509 -req -days 3650 -in my.csr -signkey my.key -out my.crt
cat my.crt my.key > certificate.pem
```


- Customize the console section in the Bare Metal service configuration file (`/etc/ironic/ironic.conf`), if you want to use SSL certificate in shellinabox, you should specify `terminal_cert_dir`. for example:

```
[console]

#
# Options defined in ironic.drivers.modules.console_utils
#

# Path to serial console terminal program. Used only by Shell
# In A Box console. (string value)
#terminal=shellinabxd

# Directory containing the terminal SSL cert (PEM) for serial
# console access. Used only by Shell In A Box console. (string
# value)
terminal_cert_dir=/tmp/ca

# Directory for holding terminal pid files. If not specified,
# the temporary directory will be used. (string value)
#terminal_pid_dir=<None>

# Time interval (in seconds) for checking the status of
# console subprocess. (integer value)
#subprocess_checking_interval=1

# Time (in seconds) to wait for the console subprocess to
# start. (integer value)
#subprocess_timeout=10
```

- Append console parameters for bare metal PXE boot in the Bare Metal service configuration file (`/etc/ironic/ironic.conf`). See the reference for configuration in [Appending kernel parameters to boot instances](#).
- Enable the `ipmitool-shellinabox` console interface, for example:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-shellinabox,no-console
```

- Configure node web console.

If the node uses a hardware type, for example `ipmi`, set the nodes console interface to `ipmitool-shellinabox`:

```
baremetal node set <node> --console-interface ipmitool-shellinabox
```

Enable the web console, for example:

```
baremetal node set <node> \
    --driver-info <terminal_port>=<customized_port>
baremetal node console enable <node>
```

Check whether the console is enabled, for example:

```
baremetal node validate <node>
```

Disable the web console, for example:

```
baremetal node console disable <node>
baremetal node unset <node> --driver-info <terminal_port>
```

The <terminal_port> is driver dependent. The actual name of this field can be checked in driver properties, for example:

```
baremetal driver property list <driver>
```

For the ipmi hardware type, this option is ipmi_terminal_port. Give a customized port number to <customized_port>, for example 8023, this customized port is used in web console url.

Get web console information for a node as follows:

```
baremetal node console show <node>
+-----+-----+
| Property      | Value                                     |
+-----+-----+
| console_enabled | True                                     |
+-----+-----+
| console_info   | {'url': u'http://<url>:<customized_port>', u'type': u
| 'shellinabox'} |
+-----+-----+
```

You can open web console using above url through web browser. If console_enabled is false, console_info is None, web console is disabled. If you want to launch web console, see the Configure node web console part.

Node serial console

Serial consoles for nodes are implemented using socat. It is supported by the ipmi and irmc hardware types.

Serial consoles can be configured in the Bare Metal service as follows:

- Install socat on the ironic conductor node. Also, socat needs to be in the \$PATH environment variable that the ironic-conductor service uses.

Installation example:

Ubuntu:

```
sudo apt-get install socat
```

RHEL8/CentOS8/Fedora:

```
sudo dnf install socat
```

- Append console parameters for bare metal PXE boot in the Bare Metal service configuration file. See the reference on how to configure them in *Appending kernel parameters to boot instances*.
- Enable the `ipmitool-socat` console interface, for example:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-socat,no-console
```

- Configure node console.

If the node uses a hardware type, for example `ipmi`, set the nodes console interface to `ipmitool-socat`:

```
baremetal node set <node> --console-interface ipmitool-socat
```

Enable the serial console, for example:

```
baremetal node set <node> --driver-info ipmi_terminal_port=<port>
baremetal node console enable <node>
```

Check whether the serial console is enabled, for example:

```
baremetal node validate <node>
```

Disable the serial console, for example:

```
baremetal node console disable <node>
baremetal node unset <node> --driver-info <ipmi_terminal_port>
```

Serial console information is available from the Bare Metal service. Get serial console information for a node from the Bare Metal service as follows:

```
baremetal node console show <node>
+-----+-----+
↪-----+
| Property      | Value                                     ↪
↪          |
+-----+-----+
↪-----+
| console_enabled | True                                     ↪
↪          |
| console_info    | {u'url': u'tcp://<host>:<port>', u'type': u'socat'} ↪
↪          |
+-----+-----+
↪-----+
```

If `console_enabled` is false or `console_info` is None then the serial console is disabled. If you want to launch serial console, see the `Configure node console`.

Node serial console of the Bare Metal service is compatible with the serial console of the Compute service. Hence, serial consoles to Bare Metal nodes can be seen and interacted with via the Dashboard

service. In order to achieve that, you need to follow the documentation for [Serial Console](#) from the Compute service.

Configuring HA

When using Bare Metal serial console under High Availability (HA) configuration, you may consider some settings below.

- If you use HAProxy, you may need to set the timeout for both client and server sides with appropriate values. Here is an example of the configuration for the timeout parameter.

```
frontend nova_serial_console
  bind 192.168.20.30:6083
  timeout client 10m # This parameter is necessary
  use_backend nova_serial_console if <...>

backend nova_serial_console
  balance source
  timeout server 10m # This parameter is necessary
  option tcpka
  option tcplog
  server controller01 192.168.30.11:6083 check inter 2000 rise 2 fall 5
  server controller02 192.168.30.12:6083 check inter 2000 rise 2 fall 5
```

- The Compute services caching feature may need to be enabled in order to make the Bare Metal serial console work under a HA configuration. Here is an example of caching configuration in `nova.conf`.

```
[cache]
enabled = true
backend = dogpile.cache.memcached
memcache_servers = memcache01:11211,memcache02:11211,memcache03:11211
```

4.2.14 Notifications

Ironic, when configured to do so, will emit notifications over a message bus that indicate different events that occur within the service. These can be consumed by any external service. Examples may include a billing or usage system, a monitoring data store, or other OpenStack services. This page describes how to enable notifications and the different kinds of notifications that Ironic may emit. The external consumer will see notifications emitted by Ironic as JSON objects structured in the following manner:

```
{
  "priority": <string, defined by the sender>,
  "event_type": <string, defined by the sender>,
  "timestamp": <string, the isotime of when the notification emitted>,
  "publisher_id": <string, defined by the sender>,
  "message_id": <uuid, generated by oslo>,
  "payload": <json serialized dict, defined by the sender>
}
```

Configuration

To enable notifications with ironic, there are two configuration options in `ironic.conf` that must be adjusted.

The first option is the `notification_level` option in the `[DEFAULT]` section of the configuration file. This can be set to `debug`, `info`, `warning`, `error`, or `critical`, and determines the minimum priority level for which notifications are emitted. For example, if the option is set to `warning`, all notifications with priority level `warning`, `error`, or `critical` are emitted, but not notifications with priority level `debug` or `info`. For information about the semantics of each log level, see the OpenStack logging standards¹. If this option is unset, no notifications will be emitted. The priority level of each available notification is documented below.

The second option is the `transport_url` option in the `[oslo_messaging_notifications]` section of the configuration. This determines the message bus used when sending notifications. If this is unset, the default transport used for RPC is used.

All notifications are emitted on the `ironic_versioned_notifications` topic in the message bus. Generally, each type of message that traverses the message bus is associated with a topic describing what the message is about. For more information, see the documentation of your chosen message bus, such as the RabbitMQ documentation².

Note that notifications may be lossy, and there's no guarantee that a notification will make it across the message bus to a consumer.

Versioning

Each notification has an associated version in the `ironic_object.version` field of the payload. Consumers are guaranteed that microversion bumps will add new fields, while macroversion bumps are backwards-incompatible and may have fields removed.

Versioned notifications are emitted by default to the `ironic_versioned_notifications` topic. This can be changed and it is configurable in the `ironic.conf` with the `versioned_notifications_topics` config option.

Available notifications

The notifications that ironic emits are described here. They are listed (alphabetically) by service first, then by `event_type`. All examples below show payloads before serialization to JSON.

ironic-api notifications

Resources CRUD notifications

These notifications are emitted from API service when ironic resources are modified as part of create, update, or delete (CRUD)³ procedures. All CRUD notifications are emitted at INFO level, except for error status that is emitted at ERROR level.

List of CRUD notifications for chassis:

¹ https://wiki.openstack.org/wiki/LoggingStandards#Log_level_definitions

² <https://www.rabbitmq.com/documentation.html>

³ https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

- baremetal.chassis.create.start
- baremetal.chassis.create.end
- baremetal.chassis.create.error
- baremetal.chassis.update.start
- baremetal.chassis.update.end
- baremetal.chassis.update.error
- baremetal.chassis.delete.start
- baremetal.chassis.delete.end
- baremetal.chassis.delete.error

Example of chassis CRUD notification:

```
{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "ChassisCRUDPayload",
    "ironic_object.version": "1.0",
    "ironic_object.data": {
      "created_at": "2016-04-10T10:13:03+00:00",
      "description": "bare 28",
      "extra": {},
      "updated_at": "2016-04-27T21:11:03+00:00",
      "uuid": "1910f669-ce8b-43c2-b1d8-cf3d65be815e"
    }
  },
  "event_type": "baremetal.chassis.update.end",
  "publisher_id": "ironic-api.hostname02"
}
```

List of CRUD notifications for deploy template:

- baremetal.deploy_template.create.start
- baremetal.deploy_template.create.end
- baremetal.deploy_template.create.error
- baremetal.deploy_template.update.start
- baremetal.deploy_template.update.end
- baremetal.deploy_template.update.error
- baremetal.deploy_template.delete.start
- baremetal.deploy_template.delete.end
- baremetal.deploy_template.delete.error

Example of deploy template CRUD notification:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "DeployTemplateCRUDPayload",
    "ironic_object.version": "1.0",
    "ironic_object.data": {
      "created_at": "2019-02-10T10:13:03+00:00",
      "extra": {},
      "name": "CUSTOM_HYPERTHREADING_ON",
      "steps": [
        {
          "interface": "bios",
          "step": "apply_configuration",
          "args": {
            "settings": [
              {
                "name": "LogicalProc",
                "value": "Enabled"
              }
            ]
          },
          "priority": 150
        }
      ],
      "updated_at": "2019-02-27T21:11:03+00:00",
      "uuid": "1910f669-ce8b-43c2-b1d8-cf3d65be815e"
    }
  },
  "event_type": "baremetal.deploy_template.update.end",
  "publisher_id": "ironic-api.hostname02"
}

```

List of CRUD notifications for node:

- baremetal.node.create.start
- baremetal.node.create.end
- baremetal.node.create.error
- baremetal.node.update.start
- baremetal.node.update.end
- baremetal.node.update.error
- baremetal.node.delete.start
- baremetal.node.delete.end
- baremetal.node.delete.error

Example of node CRUD notification:

```
{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "NodeCRUDPayload",
    "ironic_object.version": "1.13",
    "ironic_object.data": {
      "chassis_uuid": "db0eef9d-45b2-4dc0-94a8-fc283c01171f",
      "clean_step": None,
      "conductor_group": "",
      "console_enabled": False,
      "created_at": "2016-01-26T20:41:03+00:00",
      "deploy_step": None,
      "description": "my sample node",
      "driver": "ipmi",
      "driver_info": {
        "ipmi_address": "192.168.0.111",
        "ipmi_username": "root"},
      "extra": {},
      "inspection_finished_at": None,
      "inspection_started_at": None,
      "instance_info": {},
      "instance_uuid": None,
      "last_error": None,
      "lessee": None,
      "maintenance": False,
      "maintenance_reason": None,
      "fault": None,
      "bios_interface": "no-bios",
      "boot_interface": "pxe",
      "console_interface": "no-console",
      "deploy_interface": "direct",
      "inspect_interface": "no-inspect",
      "management_interface": "ipmitool",
      "network_interface": "flat",
      "power_interface": "ipmitool",
      "raid_interface": "no-raid",
      "rescue_interface": "no-rescue",
      "storage_interface": "noop",
      "vendor_interface": "no-vendor",
      "name": None,
      "owner": None,
      "power_state": "power off",
      "properties": {
        "memory_mb": 4096,
        "cpu_arch": "x86_64",
        "local_gb": 10,
        "cpus": 8},
      "protected": False,
      "protected_reason": None,
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

        "provision_state": "deploying",
        "provision_updated_at": "2016-01-27T20:41:03+00:00",
        "resource_class": None,
        "retired": None,
        "retired_reason": None,
        "target_power_state": None,
        "target_provision_state": "active",
        "traits": [
            "CUSTOM_TRAIT1",
            "HW_CPU_X86_VMX"],
        "updated_at": "2016-01-27T20:41:03+00:00",
        "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123"
    },
    "event_type": "baremetal.node.update.end",
    "publisher_id": "ironic-api.hostname02"
}

```

List of CRUD notifications for port:

- baremetal.port.create.start
- baremetal.port.create.end
- baremetal.port.create.error
- baremetal.port.update.start
- baremetal.port.update.end
- baremetal.port.update.error
- baremetal.port.delete.start
- baremetal.port.delete.end
- baremetal.port.delete.error

Example of port CRUD notification:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "PortCRUDPayload",
    "ironic_object.version": "1.3",
    "ironic_object.data": {
      "address": "77:66:23:34:11:b7",
      "created_at": "2016-02-11T15:23:03+00:00",
      "node_uuid": "5b236cab-ad4e-4220-b57c-e827e858745a",
      "extra": {},
      "is_smartnic": True,
      "local_link_connection": {},
      "physical_network": "physnet1",
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

        "portgroup_uuid": "bd2f385e-c51c-4752-82d1-7a9ec2c25f24",
        "pxe_enabled": True,
        "updated_at": "2016-03-27T20:41:03+00:00",
        "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123"
    },
    "event_type": "baremetal.port.update.end",
    "publisher_id": "ironic-api.hostname02"
}

```

List of CRUD notifications for port group:

- baremetal.portgroup.create.start
- baremetal.portgroup.create.end
- baremetal.portgroup.create.error
- baremetal.portgroup.update.start
- baremetal.portgroup.update.end
- baremetal.portgroup.update.error
- baremetal.portgroup.delete.start
- baremetal.portgroup.delete.end
- baremetal.portgroup.delete.error

Example of portgroup CRUD notification:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "PortgroupCRUDPayload",
    "ironic_object.version": "1.0",
    "ironic_object.data": {
      "address": "11:44:32:87:61:e5",
      "created_at": "2017-01-11T11:33:03+00:00",
      "node_uuid": "5b236cab-ad4e-4220-b57c-e827e858745a",
      "extra": {},
      "mode": "7",
      "name": "portgroup-node-18",
      "properties": {},
      "standalone_ports_supported": True,
      "updated_at": "2017-01-31T11:41:07+00:00",
      "uuid": "db033a40-bfed-4c84-815a-3db26bb268bb",
    }
  },
  "event_type": "baremetal.portgroup.update.end",
  "publisher_id": "ironic-api.hostname02"
}

```

List of CRUD notifications for volume connector:

- `baremetal.volumeconnector.create.start`
- `baremetal.volumeconnector.create.end`
- `baremetal.volumeconnector.create.error`
- `baremetal.volumeconnector.update.start`
- `baremetal.volumeconnector.update.end`
- `baremetal.volumeconnector.update.error`
- `baremetal.volumeconnector.delete.start`
- `baremetal.volumeconnector.delete.end`
- `baremetal.volumeconnector.delete.error`

Example of volume connector CRUD notification:

```
{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "VolumeConnectorCRUDPayload",
    "ironic_object.version": "1.0",
    "ironic_object.data": {
      "connector_id": "iqn.2017-05.org.openstack:01:d9a51732c3f",
      "created_at": "2017-05-11T05:57:36+00:00",
      "extra": {},
      "node_uuid": "4dbb4e69-99a8-4e13-b6e8-dd2ad4a20caf",
      "type": "iqn",
      "updated_at": "2017-05-11T08:28:58+00:00",
      "uuid": "19b9f3ab-4754-4725-a7a4-c43ea7e57360"
    }
  },
  "event_type": "baremetal.volumeconnector.update.end",
  "publisher_id": "ironic-api.hostname02"
}
```

List of CRUD notifications for volume target:

- `baremetal.volumetarget.create.start`
- `baremetal.volumetarget.create.end`
- `baremetal.volumetarget.create.error`
- `baremetal.volumetarget.update.start`
- `baremetal.volumetarget.update.end`
- `baremetal.volumetarget.update.error`
- `baremetal.volumetarget.delete.start`
- `baremetal.volumetarget.delete.end`
- `baremetal.volumetarget.delete.error`

Example of volume target CRUD notification:

```
{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.version": "1.0",
    "ironic_object.name": "VolumeTargetCRUDPayload"
    "ironic_object.data": {
      "boot_index": 0,
      "created_at": "2017-05-11T09:38:59+00:00",
      "extra": {},
      "node_uuid": "4dbb4e69-99a8-4e13-b6e8-dd2ad4a20caf",
      "properties": {
        "access_mode": "rw",
        "auth_method": "CHAP"
        "auth_password": "****",
        "auth_username": "urxhQCzAKr4sjyE8DivY",
        "encrypted": false,
        "qos_specs": null,
        "target_discovered": false,
        "target_iqn": "iqn.2010-10.org.openstack:volume-f0d9b0e6-b242-
↪9105-91d4-a20331693ad8",
        "target_lun": 1,
        "target_portal": "192.168.12.34:3260",
        "volume_id": "f0d9b0e6-b042-4105-91d4-a20331693ad8",
      },
      "updated_at": "2017-05-11T09:52:04+00:00",
      "uuid": "82a45833-9c58-4ec1-943c-2091ab10e47b",
      "volume_id": "f0d9b0e6-b242-9105-91d4-a20331693ad8",
      "volume_type": "iscsi"
    }
  },
  "event_type": "baremetal.volumetarget.update.end",
  "publisher_id": "ironic-api.hostname02"
}
```

Node maintenance notifications

These notifications are emitted from API service when maintenance mode is changed via API service. List of maintenance notifications for a node:

- baremetal.node.maintenance_set.start
- baremetal.node.maintenance_set.end
- baremetal.node.maintenance_set.error

start and end notifications have INFO level, error has ERROR. Example of node maintenance notification:

```
{
  "priority": "info",
```

(continues on next page)

(continued from previous page)

```
"payload":{
  "ironic_object.namespace":"ironic",
  "ironic_object.name":"NodePayload",
  "ironic_object.version":"1.15",
  "ironic_object.data":{
    "clean_step": None,
    "conductor_group": "",
    "console_enabled": False,
    "created_at": "2016-01-26T20:41:03+00:00",
    "deploy_step": None,
    "description": "my sample node",
    "driver": "ipmi",
    "extra": {},
    "inspection_finished_at": None,
    "inspection_started_at": None,
    "instance_info": {},
    "instance_uuid": None,
    "last_error": None,
    "lessee": None,
    "maintenance": True,
    "maintenance_reason": "hw upgrade",
    "fault": None,
    "bios_interface": "no-bios",
    "boot_interface": "pxe",
    "console_interface": "no-console",
    "deploy_interface": "direct",
    "inspect_interface": "no-inspect",
    "management_interface": "ipmitool",
    "network_interface": "flat",
    "power_interface": "ipmitool",
    "raid_interface": "no-raid",
    "rescue_interface": "no-rescue",
    "storage_interface": "noop",
    "vendor_interface": "no-vendor",
    "name": None,
    "owner": None,
    "power_state": "power off",
    "properties": {
      "memory_mb": 4096,
      "cpu_arch": "x86_64",
      "local_gb": 10,
      "cpus": 8},
    "protected": False,
    "protected_reason": None,
    "provision_state": "available",
    "provision_updated_at": "2016-01-27T20:41:03+00:00",
    "resource_class": None,
    "retired": None,
    "retired_reason": None,
```

(continues on next page)

(continued from previous page)

```

        "target_power_state": None,
        "target_provision_state": None,
        "traits": [
            "CUSTOM_TRAIT1",
            "HW_CPU_X86_VMX"],
        "updated_at": "2016-01-27T20:41:03+00:00",
        "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123"
    },
    "event_type": "baremetal.node.maintenance_set.start",
    "publisher_id": "ironic-api.hostname02"
}

```

ironic-conductor notifications

Node console notifications

These notifications are emitted by the ironic-conductor service when conductor service starts or stops console for the node. The notification event types for a node console are:

- baremetal.node.console_set.start
- baremetal.node.console_set.end
- baremetal.node.console_set.error
- baremetal.node.console_restore.start
- baremetal.node.console_restore.end
- baremetal.node.console_restore.error

console_set action is used when start or stop console is initiated. The console_restore action is used when the console was already enabled, but a driver must restart the console because an ironic-conductor was restarted. This may also be sent when an ironic-conductor takes over a node that was being managed by another ironic-conductor. start and end notifications have INFO level, error has ERROR. Example of node console notification:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "NodePayload",
    "ironic_object.version": "1.15",
    "ironic_object.data": {
      "clean_step": None,
      "conductor_group": "",
      "console_enabled": True,
      "created_at": "2016-01-26T20:41:03+00:00",
      "deploy_step": None,
      "description": "my sample node",
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    "driver": "ipmi",
    "extra": {},
    "inspection_finished_at": None,
    "inspection_started_at": None,
    "instance_info": {},
    "instance_uuid": None,
    "last_error": None,
    "lessee": None,
    "maintenance": False,
    "maintenance_reason": None,
    "fault": None,
    "bios_interface": "no-bios",
    "boot_interface": "pxe",
    "console_interface": "no-console",
    "deploy_interface": "direct",
    "inspect_interface": "no-inspect",
    "management_interface": "ipmitool",
    "network_interface": "flat",
    "power_interface": "ipmitool",
    "raid_interface": "no-raid",
    "rescue_interface": "no-rescue",
    "storage_interface": "noop",
    "vendor_interface": "no-vendor",
    "name": None,
    "owner": None,
    "power_state": "power off",
    "properties": {
        "memory_mb": 4096,
        "cpu_arch": "x86_64",
        "local_gb": 10,
        "cpus": 8},
    "protected": False,
    "protected_reason": None,
    "provision_state": "available",
    "provision_updated_at": "2016-01-27T20:41:03+00:00",
    "resource_class": None,
    "retired": None,
    "retired_reason": None,
    "target_power_state": None,
    "target_provision_state": None,
    "traits": [
        "CUSTOM_TRAIT1",
        "HW_CPU_X86_VMX"],
    "updated_at": "2016-01-27T20:41:03+00:00",
    "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123"
}
},
"event_type": "baremetal.node.console_set.end",
"publisher_id": "ironic-conductor.hostname01"

```

(continues on next page)

(continued from previous page)

```
}
```

baremetal.node.power_set

- `baremetal.node.power_set.start` is emitted by the `ironic-conductor` service when it begins a power state change. It has notification level `info`.
- `baremetal.node.power_set.end` is emitted when `ironic-conductor` successfully completes a power state change task. It has notification level `info`.
- `baremetal.node.power_set.error` is emitted by `ironic-conductor` when it fails to set a nodes power state. It has notification level `error`. This can occur when `ironic` fails to retrieve the old power state prior to setting the new one on the node, or when it fails to set the power state if a change is requested.

Here is an example payload for a notification with this event type. The `to_power` payload field indicates the power state to which the `ironic-conductor` is attempting to change the node:

```
{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "NodeSetPowerStatePayload",
    "ironic_object.version": "1.15",
    "ironic_object.data": {
      "clean_step": None,
      "conductor_group": "",
      "console_enabled": False,
      "created_at": "2016-01-26T20:41:03+00:00",
      "deploy_step": None,
      "description": "my sample node",
      "driver": "ipmi",
      "extra": {},
      "inspection_finished_at": None,
      "inspection_started_at": None,
      "instance_uuid": "d6ea00c1-1f94-4e95-90b3-3462d7031678",
      "last_error": None,
      "lessee": None,
      "maintenance": False,
      "maintenance_reason": None,
      "fault": None,
      "bios_interface": "no-bios",
      "boot_interface": "pxe",
      "console_interface": "no-console",
      "deploy_interface": "direct",
      "inspect_interface": "no-inspect",
      "management_interface": "ipmitool",
      "network_interface": "flat",
      "power_interface": "ipmitool",
      "raid_interface": "no-raid",
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    "rescue_interface": "no-rescue",
    "storage_interface": "noop",
    "vendor_interface": "no-vendor",
    "name": None,
    "owner": None,
    "power_state": "power off",
    "properties": {
        "memory_mb": 4096,
        "cpu_arch": "x86_64",
        "local_gb": 10,
        "cpus": 8},
    "protected": False,
    "protected_reason": None
    "provision_state": "available",
    "provision_updated_at": "2016-01-27T20:41:03+00:00",
    "resource_class": None,
    "retired": None,
    "retired_reason": None,
    "target_power_state": None,
    "target_provision_state": None,
    "traits": [
        "CUSTOM_TRAIT1",
        "HW_CPU_X86_VMX"],
    "updated_at": "2016-01-27T20:41:03+00:00",
    "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123",
    "to_power": "power on"
    },
},
"event_type": "baremetal.node.power_set.start",
"publisher_id": "ironic-conductor.hostname01"
}

```

baremetal.node.power_state_corrected

- `baremetal.node.power_state_corrected.success` is emitted by ironic-conductor when the power state on the baremetal hardware is different from the previous known power state of the node and the database is corrected to reflect this new power state. It has notification level info.

Here is an example payload for a notification with this event_type. The `from_power` payload field indicates the previous power state on the node, prior to the correction:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "NodeCorrectedPowerStatePayload",
    "ironic_object.version": "1.15",
    "ironic_object.data": {
      "clean_step": None,

```

(continues on next page)

(continued from previous page)

```
"conductor_group": "",
"console_enabled": False,
"created_at": "2016-01-26T20:41:03+00:00",
"deploy_step": None,
"description": "my sample node",
"driver": "ipmi",
"extra": {},
"inspection_finished_at": None,
"inspection_started_at": None,
"instance_uuid": "d6ea00c1-1f94-4e95-90b3-3462d7031678",
"last_error": None,
"lessee": None,
"maintenance": False,
"maintenance_reason": None,
"fault": None,
"bios_interface": "no-bios",
"boot_interface": "pxe",
"console_interface": "no-console",
"deploy_interface": "direct",
"inspect_interface": "no-inspect",
"management_interface": "ipmitool",
"network_interface": "flat",
"power_interface": "ipmitool",
"raid_interface": "no-raid",
"rescue_interface": "no-rescue",
"storage_interface": "noop",
"vendor_interface": "no-vendor",
"name": None,
"owner": None,
"power_state": "power off",
"properties": {
    "memory_mb": 4096,
    "cpu_arch": "x86_64",
    "local_gb": 10,
    "cpus": 8},
"protected": False,
"protected_reason": None,
"provision_state": "available",
"provision_updated_at": "2016-01-27T20:41:03+00:00",
"resource_class": None,
"retired": None,
"retired_reason": None,
"target_power_state": None,
"target_provision_state": None,
"traits": [
    "CUSTOM_TRAIT1",
    "HW_CPU_X86_VMX"],
"updated_at": "2016-01-27T20:41:03+00:00",
"uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123",
```

(continues on next page)

(continued from previous page)

```

        "from_power": "power on"
    },
    },
    "event_type": "baremetal.node.power_state_corrected.success",
    "publisher_id": "ironic-conductor.cond-hostname02"
}

```

baremetal.node.provision_set

- `baremetal.node.provision_set.start` is emitted by the ironic-conductor service when it begins a provision state transition. It has notification level INFO.
- `baremetal.node.provision_set.end` is emitted when ironic-conductor successfully completes a provision state transition. It has notification level INFO.
- `baremetal.node.provision_set.success` is emitted when ironic-conductor successfully changes provision state instantly, without any intermediate work required (example is AVAILABLE to MANAGEABLE). It has notification level INFO.
- `baremetal.node.provision_set.error` is emitted by ironic-conductor when it changes provision state as result of error event processing. It has notification level ERROR.

Here is an example payload for a notification with this event type. The `previous_provision_state` and `previous_target_provision_state` payload fields indicate a nodes provision states before state change, event is the FSM (finite state machine) event that triggered the state change:

```

{
  "priority": "info",
  "payload": {
    "ironic_object.namespace": "ironic",
    "ironic_object.name": "NodeSetProvisionStatePayload",
    "ironic_object.version": "1.16",
    "ironic_object.data": {
      "clean_step": None,
      "conductor_group": "",
      "console_enabled": False,
      "created_at": "2016-01-26T20:41:03+00:00",
      "deploy_step": None,
      "description": "my sample node",
      "driver": "ipmi",
      "driver_internal_info": {
        "is_whole_disk_image": True},
      "extra": {},
      "inspection_finished_at": None,
      "inspection_started_at": None,
      "instance_info": {},
      "instance_uuid": None,
      "last_error": None,
      "lessee": None,
      "maintenance": False,

```

(continues on next page)

(continued from previous page)

```
    "maintenance_reason": None,
    "fault": None,
    "bios_interface": "no-bios",
    "boot_interface": "pxe",
    "console_interface": "no-console",
    "deploy_interface": "direct",
    "inspect_interface": "no-inspect",
    "management_interface": "ipmitool",
    "network_interface": "flat",
    "power_interface": "ipmitool",
    "raid_interface": "no-raid",
    "rescue_interface": "no-rescue",
    "storage_interface": "noop",
    "vendor_interface": "no-vendor",
    "name": None,
    "owner": None,
    "power_state": "power off",
    "properties": {
        "memory_mb": 4096,
        "cpu_arch": "x86_64",
        "local_gb": 10,
        "cpus": 8},
    "protected": False,
    "protected_reason": None,
    "provision_state": "deploying",
    "provision_updated_at": "2016-01-27T20:41:03+00:00",
    "resource_class": None,
    "retired": None,
    "retired_reason": None,
    "target_power_state": None,
    "target_provision_state": "active",
    "traits": [
        "CUSTOM_TRAIT1",
        "HW_CPU_X86_VMX"],
    "updated_at": "2016-01-27T20:41:03+00:00",
    "uuid": "1be26c0b-03f2-4d2e-ae87-c02d7f33c123",
    "previous_provision_state": "available",
    "previous_target_provision_state": None,
    "event": "deploy"
    },
    "event_type": "baremetal.node.provision_set.start",
    "publisher_id": "ironic-conductor.hostname01"
}
```

4.2.15 Conductor Groups

Overview

Large scale operators tend to have needs that involve creating well defined and delineated resources. In some cases, these systems may reside close by or in far away locations. Reasoning may be simple or complex, and yet is only known to the deployer and operator of the infrastructure.

A common case is the need for delineated high availability domains where it would be much more efficient to manage a datacenter in Antarctica with a conductor in Antarctica, as opposed to a conductor in New York City.

How it works

Starting in ironic 11.1, each node has a `conductor_group` field which influences how the ironic conductor calculates (and thus allocates) baremetal nodes under ironics management. This calculation is performed independently by each operating conductor and as such if a conductor has a `[conductor]conductor_group` configuration option defined in its `ironic.conf` configuration file, the conductor will then be limited to only managing nodes with a matching `conductor_group` string.

Note: Any conductor without a `[conductor]conductor_group` setting will only manage baremetal nodes without a `conductor_group` value set upon node creation. If no such conductor is present when conductor groups are configured, node creation will fail unless a `conductor_group` is specified upon node creation.

Warning: Nodes without a `conductor_group` setting can only be managed when a conductor exists that does not have a `[conductor]conductor_group` defined. If all conductors have been migrated to use a conductor group, such nodes are effectively orphaned.

How to use

A conductor group value may be any case insensitive string up to 255 characters long which matches the `^[a-zA-Z0-9_-\.\]*$` regular expression.

1. Set the `[conductor]conductor_group` option in `ironic.conf` on one or more, but not all conductors:

```
[conductor]
conductor_group = OperatorDefinedString
```

2. Restart the `ironic-conductor` service.
3. Set the conductor group on one or more nodes:

```
baremetal node set \
  --conductor-group "OperatorDefinedString" <uuid>
```

4. As desired and as needed, remaining conductors can be updated with the first two steps. Please be mindful of the constraints covered earlier in the document related to ability to manage nodes.

4.2.16 Security Overview

While the Bare Metal service is intended to be a secure application, it is important to understand what it does and does not cover today.

Deployers must properly evaluate their use case and take the appropriate actions to secure their environment(s). This document is intended to provide an overview of what risks an operator of the Bare Metal service should be aware of. It is not intended as a How-To guide for securing a data center or an OpenStack deployment.

REST API: user roles and policy settings

Warning: This information is presently in flux as of the Wallaby release with the implementation of Secure RBAC where system and project scoped requests are able to be parsed and default access controls support a delineation of roles and responsibilities through the roles. Please see [Secure RBAC](#).

Beginning with the Newton (6.1.0) release, the Bare Metal service allows operators significant control over API access:

- Access may be restricted to each method (GET, PUT, etc) for each REST resource. Defaults are provided with the release and defined in code.
- Access may be divided between an administrative role with full access and observer role with read-only access. By default, these roles are assigned the names `baremetal_admin` and `baremetal_observer`, respectively.
- By default, passwords and instance secrets are hidden in `driver_info` and `instance_info`, respectively. In case of debugging or diagnosing, the behavior can be overridden by changing the policy file. To allow password in `driver_info` unmasked for users with administrative privileges, apply following changes to policy configuration file:

```
"show_password": "rule:is_admin"
```

And restart the Bare Metal API service to take effect. Please check [Policies](#) for more details.

Prior to the Newton (6.1.0) release, the Bare Metal service only supported two policy options:

- API access may be secured by a simple policy rule: users with administrative privileges may access all API resources, whereas users without administrative privileges may only access public API resources.
- Passwords contained in the `driver_info` field may be hidden from all API responses with the `show_password` policy setting. This defaults to always hide passwords, regardless of the users role. You can override it with policy configuration as described above.

Multi-tenancy

There are two aspects of multitenancy to consider when evaluating a deployment of the Bare Metal Service: interactions between tenants on the network, and actions one tenant can take on a machine that will affect the next tenant.

Network Interactions

Interactions between tenants workloads running simultaneously on separate servers include, but are not limited to: IP spoofing, packet sniffing, and network man-in-the-middle attacks.

By default, the Bare Metal service provisions all nodes on a flat network, and does not take any precautions to avoid or prevent interaction between tenants. This can be addressed by integration with the OpenStack Identity, Compute, and Networking services, so as to provide tenant-network isolation. Additional documentation on [network multi-tenancy](#) is available.

Lingering Effects

Interactions between tenants placed sequentially on the same server include, but are not limited to: changes in BIOS settings, modifications to firmware, or files left on disk or peripheral storage devices (if these devices are not erased between uses).

By default, the Bare Metal service will erase (clean) the local disk drives during the cleaning phase, after deleting an instance. It *does not* reset BIOS or reflash firmware or peripheral devices. This can be addressed through customizing the utility ramdisk used during the cleaning phase. See details in the [Firmware security](#) section.

Firmware security

When the Bare Metal service deploys an operating system image to a server, that image is run natively on the server without virtualization. Any user with administrative access to the deployed instance has administrative access to the underlying hardware.

Most servers default settings do not prevent a privileged local user from gaining direct access to hardware devices. Such a user could modify device or firmware settings, and potentially flash new firmware to the device, before deleting their instance and allowing the server to be allocated to another user.

If the `[conductor]/automated_clean` configuration option is enabled (and the `[deploy]/erase_devices_priority` configuration option is not zero), the Bare Metal service will securely erase all local disk devices within a machine during instance deletion. However, the service does not ship with any code that will validate the integrity of, or make any modifications to, system or device firmware or firmware settings.

Operators are encouraged to write their own hardware manager plugins for the `ironic-python-agent` ramdisk. This should include custom `clean` steps that would be run during the *Node cleaning* process, as part of Node de-provisioning. The `clean` steps would perform the specific actions necessary within that environment to ensure the integrity of each servers firmware.

Ideally, an operator would work with their hardware vendor to ensure that proper firmware security measures are put in place ahead of time. This could include:

- installing signed firmware for BIOS and peripheral devices

- using a TPM (Trusted Platform Module) to validate signatures at boot time
- booting machines in *UEFI secure boot mode*, rather than BIOS mode, to validate kernel signatures
- disabling local (in-band) access from the host OS to the management controller (BMC)
- disabling modifications to boot settings from the host OS

Additional references:

- *Node cleaning*

UEFI secure boot mode

Some hardware types support turning *UEFI secure boot* dynamically when deploying an instance. Currently these are *iLO driver*, *iRMC driver* and *Redfish driver*.

Support for the UEFI secure boot is declared by adding the `secure_boot` capability in the `capabilities` parameter in the `properties` field of a node. `secure_boot` is a boolean parameter and takes value as `true` or `false`.

To enable `secure_boot` on a node add it to `capabilities`:

```
baremetal node set <node> --property capabilities='secure_boot:true'
```

Alternatively use *Hardware Inspection* to automatically populate the secure boot capability.

Warning: UEFI secure boot only works in UEFI boot mode, see *Boot mode support* for how to turn it on and off.

Compatible images

Use element `ubuntu-signed` or `fedora` to build signed deploy ISO and user images with `diskimage-builder`.

The below command creates files named `cloud-image-boot.iso`, `cloud-image.initrd`, `cloud-image.vmlinuz` and `cloud-image.qcow2` in the current working directory:

```
cd <path-to-diskimage-builder>  
./bin/disk-image-create -o cloud-image ubuntu-signed baremetal iso
```

Ensure the public key of the signed image is loaded into bare metal to deploy signed images.

Enabling with OpenStack Compute

Nodes having `secure_boot` set to `true` may be requested by adding an `extra_spec` to the nova flavor:

```
openstack flavor set <flavor> --property capabilities:secure_boot="true"  
openstack server create --flavor <flavor> --image <image> instance-1
```


If `capabilities` is used in `extra_spec` as above, nova scheduler (`ComputeCapabilitiesFilter`) will match only ironic nodes which have the `secure_boot` set appropriately in `properties/capabilities`. It will filter out rest of the nodes.

The above facility for matching in nova can be used in heterogeneous environments where there is a mix of machines supporting and not supporting UEFI secure boot, and operator wants to provide a choice to the user regarding secure boot. If the flavor doesn't contain `secure_boot` then nova scheduler will not consider secure boot mode as a placement criteria, hence user may get a secure boot capable machine that matches with user specified flavors but deployment would not use its secure boot capability. Secure boot deploy would happen only when it is explicitly specified through flavor.

Enabling standalone

To request secure boot for an instance in standalone mode (without OpenStack Compute), you need to add the capability directly to the nodes `instance_info`:

```
baremetal node set <node> --instance-info capabilities='{"secure_boot": "true
↪"}'
```

Other considerations

Internal networks

Access to networks which the Bare Metal service uses internally should be prohibited from outside. These networks are the ones used for management (with the nodes BMC controllers), provisioning, cleaning (if used) and rescuing (if used).

This can be done with physical or logical network isolation, traffic filtering, etc.

Management interface technologies

Some nodes support more than one management interface technology (vendor and IPMI for example). If you use only one modern technology for out-of-band node access, it is recommended that you disable IPMI since the IPMI protocol is not secure. If IPMI is enabled, in most cases a local OS administrator is able to work in-band with IPMI settings without specifying any credentials, as this is a DCMI specification requirement.

Tenant network isolation

If you use tenant network isolation, services (TFTP or HTTP) that handle the nodes boot files should serve requests only from the internal networks that are used for the nodes being deployed and cleaned.

TFTP protocol does not support per-user access control at all.

For HTTP, there is no generic and safe way to transfer credentials to the node.

Also, tenant network isolation is not intended to work with network-booting a node by default, once the node has been provisioned.

API endpoints for RAM disk use

There are *two (unauthorized) endpoints* in the Bare Metal API that are intended for use by the ironic-python-agent RAM disk. They are not intended for public use.

These endpoints can potentially cause security issues. Access to these endpoints from external or untrusted networks should be prohibited. An easy way to do this is to:

- set up two groups of API services: one for external requests, the second for deploy RAM disks requests.
- to disable unauthorized access to these endpoints in the (first) API services group that serves external requests, the following lines should be added to the `policy.yaml` file:

```
# Send heartbeats from IPA ramdisk
"baremetal:node:ipa_heartbeat": "rule:is_admin"

# Access IPA ramdisk functions
"baremetal:driver:ipa_lookup": "rule:is_admin"
```

4.2.17 Troubleshooting Ironic

Nova returns No valid host was found Error

Sometimes Nova Conductor log file `nova-conductor.log` or a message returned from Nova API contains the following error:

```
NoValidHost: No valid host was found. There are not enough hosts available.
```

No valid host was found means that the Nova Scheduler could not find a bare metal node suitable for booting the new instance.

This in turn usually means some mismatch between resources that Nova expects to find and resources that Ironic advertised to Nova.

A few things should be checked in this case:

1. Make sure that enough nodes are in `available` state, not in `maintenance` mode and not already used by an existing instance. Check with the following command:

```
baremetal node list --provision-state available --no-maintenance --
↪ unassociated
```

If this command does not show enough nodes, use generic `baremetal node list` to check other nodes. For example, nodes in `manageable` state should be made available:

```
baremetal node provide <IRONIC NODE>
```

The Bare metal service automatically puts a node in `maintenance` mode if there are issues with accessing its management interface. Check the power credentials (e.g. `ipmi_address`, `ipmi_username` and `ipmi_password`) and then move the node out of `maintenance` mode:

```
baremetal node maintenance unset <IRONIC NODE>
```

The `node validate` command can be used to verify that all required fields are present. The following command should not return anything:

```
baremetal node validate <IRONIC NODE> | grep -E '(power|management)\
↳W*False'
```

Maintenance mode will be also set on a node if automated cleaning has failed for it previously.

2. Make sure that you have Compute services running and enabled:

```
$ openstack compute service list --service nova-compute
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+
| ID | Binary          | Host          | Zone | Status | State | Updated At |
↳+-----+-----+-----+-----+-----+-----+-----+
↳+-----+
| 7 | nova-compute | example.com | nova | enabled | up    | 2017-09-
↳04T13:14:03.000000 |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+
```

By default, a Compute service is disabled after 10 consecutive build failures on it. This is to ensure that new build requests are not routed to a broken Compute service. If it is the case, make sure to fix the source of the failures, then re-enable it:

```
openstack compute service set --enable <COMPUTE HOST> nova-compute
```

3. Starting with the Pike release, check that all your nodes have the `resource_class` field set using the following command:

```
baremetal node list --fields uuid name resource_class
```

Then check that the flavor(s) are configured to request these resource classes via their properties:

```
openstack flavor show <FLAVOR NAME> -f value -c properties
```

For example, if your node has resource class `baremetal-large`, it will be matched by a flavor with property `resources:CUSTOM_BAREMETAL_LARGE` set to 1. See [Create flavors for use with the Bare Metal service](#) for more details on the correct configuration.

4. Upon scheduling, Nova will query the Placement API service for the available resource providers (in the case of Ironic: nodes with a given resource class). If placement does not have any allocation candidates for the requested resource class, the request will result in a No valid host was found error. It is hence sensible to check if Placement is aware of resource providers (nodes) for the requested resource class with:

```
$ openstack allocation candidate list --resource CUSTOM_BAREMETAL_LARGE='1
↳'
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+
| # | allocation          | resource provider |
↳+-----+-----+-----+-----+-----+-----+-----+
↳| inventory used/capacity |
```

(continues on next page)

(continued from previous page)

```

+---+-----+-----+-----+-----+-----+-----+-----+-----+
↪+-----+
| 1 | CUSTOM_BAREMETAL_LARGE=1 | 2f7b9c69-c1df-4e40-b94e-5821a4ea0453 |
↪| CUSTOM_BAREMETAL_LARGE=0/1 |
+---+-----+-----+-----+-----+-----+-----+-----+-----+
↪+-----+

```

For Ironic, the resource provider is the UUID of the available Ironic node. If this command returns an empty list (or does not contain the targeted resource provider), the operator needs to understand first, why the resource tracker has not reported this provider to placement. Potential explanations include:

- the resource tracker cycle has not finished yet and the resource provider will appear once it has (the time to finish the cycle scales linearly with the number of nodes the corresponding nova-compute service manages);
 - the node is in a state where the resource tracker does not consider it to be eligible for scheduling, e.g. when the node has maintenance set to True; make sure the target nodes are in available and maintenance is False;
5. If you do not use scheduling based on resource classes, then the nodes properties must have been set either manually or via inspection. For each node with available state check that the properties JSON field has valid values for the keys cpus, cpu_arch, memory_mb and local_gb. Example of valid properties:

```

$ baremetal node show <IRONIC NODE> --fields properties
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+
| Property | Value |
↪ |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+
| properties | {u'memory_mb': u'8192', u'cpu_arch': u'x86_64', u'local_gb
↪': u'41', u'cpus': u'4'} |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+

```

Warning: If you're using exact match filters in the Nova Scheduler, make sure the flavor and the node properties match exactly.

6. The Nova flavor that you are using does not match any properties of the available Ironic nodes. Use

```
openstack flavor show <FLAVOR NAME>
```

to compare. The extra specs in your flavor starting with capability: should match ones in node.properties['capabilities'].

Note: The format of capabilities is different in Nova and Ironic. E.g. in Nova flavor:

```
$ openstack flavor show <FLAVOR NAME> -c properties
+-----+-----+
| Field      | Value                                |
+-----+-----+
| properties | capabilities:boot_option='local' |
+-----+-----+
```

But in Ironic node:

```
$ baremetal node show <IRONIC NODE> --fields properties
+-----+-----+
| Property   | Value                                |
+-----+-----+
| properties | {u'capabilities': u'boot_option:local'} |
+-----+-----+
```

7. After making changes to nodes in Ironic, it takes time for those changes to propagate from Ironic to Nova. Check that

```
openstack hypervisor stats show
```

correctly shows total amount of resources in your system. You can also check `openstack hypervisor show <IRONIC NODE>` to see the status of individual Ironic nodes as reported to Nova.

8. Figure out which Nova Scheduler filter ruled out your nodes. Check the `nova-scheduler` logs for lines containing something like:

```
Filter ComputeCapabilitiesFilter returned 0 hosts
```

The name of the filter that removed the last hosts may give some hints on what exactly was not matched. See [Nova filters documentation](#) for more details.

9. If none of the above helped, check Ironic conductor log carefully to see if there are any conductor-related errors which are the root cause for No valid host was found. If there are any Error in deploy of node <IRONIC-NODE-UUID>: [Errno 28] error messages in Ironic conductor log, it means the conductor run into a special error during deployment. So you can check the log carefully to fix or work around and then try again.

Patching the Deploy Ramdisk

When debugging a problem with deployment and/or inspection you may want to quickly apply a change to the ramdisk to see if it helps. Of course you can inject your code and/or SSH keys during the ramdisk build (depends on how exactly you've built your ramdisk). But it's also possible to quickly modify an already built ramdisk.

Create an empty directory and unpack the ramdisk content there:

```
$ mkdir unpack
$ cd unpack
$ gzip -dc /path/to/the/ramdisk | cpio -id
```

The last command will result in the whole Linux file system tree unpacked in the current directory. Now you can modify any files you want. The actual location of the files will depend on the way you've built the ramdisk.

Note: On a systemd-based system you can use the `systemd-nspawn` tool (from the `systemd-container` package) to create a lightweight container from the unpacked filesystem tree:

```
$ sudo systemd-nspawn --directory /path/to/unpacked/ramdisk/ /bin/bash
```

This will allow you to run commands within the filesystem, e.g. use package manager. If the ramdisk is also systemd-based, and you have login credentials set up, you can even boot a real ramdisk environment with

```
$ sudo systemd-nspawn --directory /path/to/unpacked/ramdisk/ --boot
```

After you've done the modifications, pack the whole content of the current directory back:

```
$ find . | cpio -H newc -o | gzip -c > /path/to/the/new/ramdisk
```

Note: You don't need to modify the kernel (e.g. `tinyipa-master.vmlinuz`), only the ramdisk part.

API Errors

The `debug_tracebacks_in_api` config option may be set to return tracebacks in the API response for all 4xx and 5xx errors.

Retrieving logs from the deploy ramdisk

When troubleshooting deployments (specially in case of a deploy failure) it's important to have access to the deploy ramdisk logs to be able to identify the source of the problem. By default, Ironic will retrieve the logs from the deploy ramdisk when the deployment fails and save it on the local filesystem at `/var/log/ironic/deploy`.

To change this behavior, operators can make the following changes to `/etc/ironic/ironic.conf` under the `[agent]` group:

- `deploy_logs_collect`: Whether Ironic should collect the deployment logs on deployment. Valid values for this option are:
 - `on_failure` (**default**): Retrieve the deployment logs upon a deployment failure.
 - `always`: Always retrieve the deployment logs, even if the deployment succeeds.
 - `never`: Disable retrieving the deployment logs.
- `deploy_logs_storage_backend`: The name of the storage backend where the logs will be stored. Valid values for this option are:
 - `local` (**default**): Store the logs in the local filesystem.
 - `swift`: Store the logs in Swift.

- `deploy_logs_local_path`: The path to the directory where the logs should be stored, used when the `deploy_logs_storage_backend` is configured to `local`. By default logs will be stored at `/var/log/ironic/deploy`.
- `deploy_logs_swift_container`: The name of the Swift container to store the logs, used when the `deploy_logs_storage_backend` is configured to `swift`. By default `ironic_deploy_logs_container`.
- `deploy_logs_swift_days_to_expire`: Number of days before a log object is marked as expired in Swift. If `None`, the logs will be kept forever or until manually deleted. Used when the `deploy_logs_storage_backend` is configured to `swift`. By default **30** days.

When the logs are collected, Ironic will store a `tar.gz` file containing all the logs according to the `deploy_logs_storage_backend` configuration option. All log objects will be named with the following pattern:

```
<node>[_<instance-uuid>]_<timestamp yyyy-mm-dd-hh:mm:ss>.tar.gz
```

Note: The `instance_uuid` field is not required for deploying a node when Ironic is configured to be used in standalone mode. If present it will be appended to the name.

Accessing the log data

When storing in the local filesystem

When storing the logs in the local filesystem, the log files can be found at the path configured in the `deploy_logs_local_path` configuration option. For example, to find the logs from the node `5e9258c4-cfda-40b6-86e2-e192f523d668`:

```
$ ls /var/log/ironic/deploy | grep 5e9258c4-cfda-40b6-86e2-e192f523d668
5e9258c4-cfda-40b6-86e2-e192f523d668_88595d8a-6725-4471-8cd5-c0f3106b6898_
↪2016-08-08-13:52:12.tar.gz
5e9258c4-cfda-40b6-86e2-e192f523d668_db87f2c5-7a9a-48c2-9a76-604287257c1b_
↪2016-08-08-14:07:25.tar.gz
```

Note: When saving the logs to the filesystem, operators may want to enable some form of rotation for the logs to avoid disk space problems.

When storing in Swift

When using Swift, operators can associate the objects in the container with the nodes in Ironic and search for the logs for the node `5e9258c4-cfda-40b6-86e2-e192f523d668` using the `prefix` parameter. For example:

```
$ swift list ironic_deploy_logs_container -p 5e9258c4-cfda-40b6-86e2-
↪e192f523d668
5e9258c4-cfda-40b6-86e2-e192f523d668_88595d8a-6725-4471-8cd5-c0f3106b6898_
↪2016-08-08-13:52:12.tar.gz
5e9258c4-cfda-40b6-86e2-e192f523d668_db87f2c5-7a9a-48c2-9a76-604287257c1b_
↪2016-08-08-14:07:25.tar.gz
```

To download a specific log from Swift, do:

```
$ swift download ironic_deploy_logs_container "5e9258c4-cfda-40b6-86e2-
↪e192f523d668_db87f2c5-7a9a-48c2-9a76-604287257c1b_2016-08-08-14:07:25.tar.gz
↪"
5e9258c4-cfda-40b6-86e2-e192f523d668_db87f2c5-7a9a-48c2-9a76-604287257c1b_
↪2016-08-08-14:07:25.tar.gz [auth 0.341s, headers 0.391s, total 0.391s, 0.
↪531 MB/s]
```

The contents of the log file

The log is just a `.tar.gz` file that can be extracted as:

```
$ tar xvf <file path>
```

The contents of the file may differ slightly depending on the distribution that the deploy ramdisk is using:

- For distributions using `systemd` there will be a file called **journal** which contains all the system logs collected via the `journalctl` command.
- For other distributions, the ramdisk will collect all the contents of the `/var/log` directory.

For all distributions, the log file will also contain the output of the following commands (if present): `ps`, `df`, `ip addr` and `iptables`.

Here's one example when extracting the content of a log file for a distribution that uses `systemd`:

```
$ tar xvf 5e9258c4-cfda-40b6-86e2-e192f523d668_88595d8a-6725-4471-8cd5-
↪c0f3106b6898_2016-08-08-13:52:12.tar.gz
df
ps
journal
ip_addr
iptables
```


DHCP during PXE or iPXE is inconsistent or unreliable

This can be caused by the spanning tree protocol delay on some switches. The delay prevents the switch port moving to forwarding mode during the nodes attempts to PXE, so the packets never make it to the DHCP server. To resolve this issue you should set the switch port that connects to your baremetal nodes as an edge or PortFast type port. Configured in this way the switch port will move to forwarding mode as soon as the link is established. An example on how to do that for a Cisco Nexus switch is:

```
$ config terminal
$ (config) interface eth1/11
$ (config-if) spanning-tree port type edge
```

Why does X issue occur when I am using LACP bonding with iPXE?

If you are using iPXE, an unfortunate aspect of its design and interaction with networking is an automatic response as a Link Aggregation Control Protocol (or LACP) peer to remote switches. iPXE does this for only the single port which is used for network booting.

In theory, this may help establish the port link-state faster with some switch vendors, but the official reasoning as far as the Ironic Developers are aware is not documented for iPXE. The end result of this is that once iPXE has stopped responding to LACP messages from the peer port, which occurs as part of the process of booting a ramdisk and iPXE handing over control to a full operating-system, switches typically begin a timer to determine how to handle the failure. This is because, depending on the mode of LACP, this can be interpreted as a switch or network fabric failure.

This may demonstrate as any number of behaviors or issues from ramdisks finding they are unable to acquire DHCP addresses over the network interface to downloads abruptly stalling, to even minor issues such as LLDP port data being unavailable in introspection.

Overall:

- Ironics agent doesnt officially support LACP and the Ironic community generally believes this may cause more problems than it would solve. During the Victoria development cycle, we added retry logic for most actions in an attempt to navigate the worst-known default hold-down timers to help ensure a deployment does not fail due to a short-lived transitory network connectivity failure in the form of a switch port having moved to a temporary blocking state. Where applicable and possible, many of these patches have been backported to supported releases. These patches also require that the switchport has an eventual fallback to a non-bonded mode. If the port remains in a blocking state, then traffic will be unable to flow and the deployment is likely to time out.
- If you must use LACP, consider *passive* LACP negotiation settings in the network switch as opposed to *active*. The difference being with passive the connected workload is likely a server where it should likely request the switch to establish the Link Aggregate. This is instead of being treated as if its possibly another switch.
- Consult your switch vendors support forums. Some vendors have recommended port settings for booting machines using iPXE with their switches.

IPMI errors

When working with IPMI, several settings need to be enabled depending on vendors.

Enable IPMI over LAN

Machines may not have IPMI access over LAN enabled by default. This could cause the IPMI port to be unreachable through `ipmitool`, as shown:

```
$ ipmitool -I lan -H ipmi_host -U ipmi_user -P ipmi_pass chassis power status
Error: Unable to establish LAN session
```

To fix this, enable *IPMI over lan* setting using your BMC tool or web app.

Troubleshooting lanplus interface

When working with lanplus interfaces, you may encounter the following error:

```
$ ipmitool -I lanplus -H ipmi_host -U ipmi_user -P ipmi_pass power status
Error in open session response message : insufficient resources for session
Error: Unable to establish IPMI v2 / RMCP+ session
```

To fix that issue, please enable *RMCP+ Cipher Suite3 Configuration* setting using your BMC tool or web app.

Why are my nodes stuck in a -ing state?

The Ironic conductor uses states ending with `ing` as a signifier that the conductor is actively working on something related to the node.

Often, this means there is an internal lock or reservation set on the node and the conductor is downloading, uploading, or attempting to perform some sort of Input/Output operation.

In the case the conductor gets stuck, these operations should timeout, but there are cases in operating systems where operations are blocked until completion. These sorts of operations can vary based on the specific environment and operating configuration.

What can cause these sorts of failures?

Typical causes of such failures are going to be largely rooted in the concept of `iowait`, either in the form of downloading from a remote host or reading or writing to the disk of the conductor. An operator can use the `iostat` tool to identify the percentage of CPU time spent waiting on storage devices.

The fields that will be particularly important are the `iowait`, `await`, and `tps` ones, which can be read about in the `iostat` manual page.

In the case of network file systems, for backing components such as image caches or distributed `tftpboot` or `httpboot` folders, IO operations failing on these can, depending on operating system and underlying client settings, cause threads to be stuck in a blocking wait state, which is realistically undetectable short the operating system logging connectivity errors or even lock manager access errors.

For example with `nfs`, the underlying client recovery behavior, in terms of `soft`, `hard`, `softreval`, `nosoftreval`, will largely impact this behavior, but also NFS server settings can impact this behavior. A solid sign that this is a failure, is when an `ls /path/to/nfs` command hangs for a period of time. In such cases, the Storage Administrator should be consulted and network connectivity investigated for errors before trying to recover to proceed.

The bad news for IO related failures

If the node has a populated `reservation` field, and has not timed out or proceeded to a `fail` state, then the conductor process will likely need to be restarted. This is because the worker thread is hung with-in the conductor.

Manual intervention with-in Ironics database is *not* advised to try and un-wedge the machine in this state, and restarting the conductor is encouraged.

Note: Ironics conductor, upon restart, clears reservations for nodes which were previously managed by the conductor before restart.

If a distributed or network file system is in use, it is highly recommended that the operating system of the node running the conductor be rebooted as the running conductor may not even be able to exit in the state of an IO failure, again dependent upon site and server configuration.

File Size != Disk Size

An easy to make misconception is that a 2.4 GB file means that only 2.4 GB is written to disk. But if that files virtual size is 20 GB, or 100 GB things can become very problematic and extend the amount of time the node spends in `deploying` and `deploy wait` states.

Again, these sorts of cases will depend upon the exact configuration of the deployment, but hopefully these are areas where these actions can occur.

- Conversion to raw image files upon download to the conductor, from the `[DEFAULT]force_raw_images` option. Users using Glance may also experience issues here as the conductor will cache the image to be written which takes place when the `[agent]image_download_source` is set to `http` instead of `swift`.

Note: The QCOW2 image conversion utility does consume quite a bit of memory when converting images or writing them to the end storage device. This is because the files are not sequential in nature, and must be re-assembled from an internal block mapping. Internally Ironic limits this to 1GB of RAM. Operators performing large numbers of deployments may wish to disable raw images in these sorts of cases in order to minimize the conductor becoming a limiting factor due to memory and network IO.

Why are my nodes stuck in a wait state?

The Ironic conductor uses states containing `wait` as a signifier that the conductor is waiting for a callback from another component, such as the Ironic Python Agent or the Inspector. If this feedback does not arrive, the conductor will time out and the node will eventually move to a `failed` state. Depending on the configuration and the circumstances, however, a node can stay in a `wait` state for a long time or even never time out. The list of such wait states includes:

- `clean wait` for cleaning,
- `inspect wait` for introspection,
- `rescue wait` for rescuing, and
- `wait call-back` for deploying.

Communication issues between the conductor and the node

One of the most common issues when nodes seem to be stuck in a wait state occur when the node never received any instructions or does not react as expected: the conductor moved the node to a wait state but the node will never call back. Examples include wrong ciphers which will make `ipmitool` get stuck or BMCs in a state where they accept commands, but don't do the requested task (or only a part of it, like shutting off, but not starting). It is useful in these cases to see via a ping or the console if and which action the node is performing. If the node does not seem to react to the requests sent by the conductor, it may be worthwhile to try the corresponding action out-of-band, e.g. confirm that power on/off commands work when directly sent to the BMC. The section on *IPMI errors* above gives some additional points to check. In some situations, a BMC reset may be necessary.

Ironic Python Agent stuck

Nodes can also get remain in a wait state when the component the conductor is waiting for gets stuck, e.g. when a hardware manager enters a loop or is waiting for an event that is never happening. In these cases, it might be helpful to connect to the IPA and inspect its logs, see the trouble shooting guide of the [ironic-python-agent \(IPA\)](#) on how to do this.

Deployments fail with failed to update MAC address

The design of the integration with the Networking service (`neutron`) is such that once virtual ports have been created in the API, their MAC address must be updated in order for the DHCP server to be able to appropriately reply.

This can sometimes result in errors being raised indicating that the MAC address is already in use. This is because at some point in the past, a virtual interface was orphaned either by accident or by some unexpected glitch, and a previous entry is still present in Neutron.

This error looks something like this when reported in the `ironic-conductor` log output.:

```
Failed to update MAC address on Neutron port 305beda7-0dd0-4fec-b4d2-78b7aa4e8e6a.:
MacAddressInUseClient: Unable to complete operation for network 1e252627-6223-4076-
a2b9-6f56493c9bac. The mac address 52:54:00:7c:c4:56 is in use.
```

Because we have no idea about this entry, we fail the deployment process as we can't make a number of assumptions in order to attempt to automatically resolve the conflict.

How did I get here?

Originally this was a fairly easy issue to encounter. The retry logic path which resulted between the Orchestration (heat) and Compute (nova) services, could sometimes result in additional un-necessary ports being created.

Bugs of this class have been largely resolved since the Rocky development cycle. Since then, the way this can become encountered is due to Networking (neutron) VIF attachments not being removed or deleted prior to deleting a port in the Bare Metal service.

Ultimately, the key of this is that the port is being deleted. Under most operating circumstances, there really is no need to delete the port, and VIF attachments are stored on the port object, so deleting the port CAN result in the VIF not being cleaned up from Neutron.

Under normal circumstances, when deleting ports, a node should be in a stable state, and the node should not be provisioned. If the `baremetal port delete` command fails, this may indicate that a known VIF is still attached. Generally if they are transitory from cleaning, provisioning, rescuing, or even inspection, getting the node to the `available` state will unblock your delete operation, that is unless there is a tenant VIF attachment. In that case, the vif will need to be removed from within the Bare Metal service using the `baremetal node vif detach` command.

A port can also be checked to see if there is a VIF attachment by consulting the ports `internal_info` field.

Warning: The `maintenance` flag can be used to force the nodes port to be deleted, however this will disable any check that would normally block the user from issuing a delete and accidentally orphaning the VIF attachment record.

How do I resolve this?

Generally, you need to identify the port with the offending MAC address. Example:

```
openstack port list mac-address 52:54:00:7c:c4:56
```

From the commands output, you should be able to identify the `id` field. Using that, you can delete the port. Example:

```
openstack port delete <id>
```

Warning: Before deleting a port, you should always verify that it is no longer in use or no longer seems applicable/operable. If multiple deployments of the Bare Metal service with a single Neutron, the possibility that an inventory typo, or possibly even a duplicate MAC address exists, which could also produce the same basic error message.

My test VM image does not deploy mount point does not exist

What is likely occurring

The image attempting to be deployed likely is a partition image where the file system that the user wishes to boot from lacks the required folders, such as `/dev` and `/proc`, which are required to install a bootloader for a Linux OS image

It should be noted that similar errors can also occur with whole disk images where we are attempting to setup the UEFI bootloader configuration. That being said, in this case, the image is likely invalid or contains an unexpected internal structure.

Users performing testing may choose something that they believe will work based on it working for virtual machines. These images are often attractive for testing as they are generic and include basic support for establishing networking and possibly installing user keys. Unfortunately, these images often lack drivers and firmware required for many different types of physical hardware which makes using them very problematic. Additionally, images such as [Cirros](#) do not have any contents in the root filesystem (i.e. an empty filesystem), as they are designed for the `ramdisk` to write the contents to disk upon boot.

How do I not encounter this issue?

We generally recommend using [diskimage-builder](#) or vendor supplied images. Centos, Ubuntu, Fedora, and Debian all publish operating system images which do generally include drivers and firmware for physical hardware. Many of these published cloud images, also support auto-configuration of networking AND population of user keys.

Issues with autoconfigured TLS

These issues will manifest as an error in `ironic-conductor` logs looking similar to (lines are wrapped for readability):

```
ERROR ironic.drivers.modules.agent_client [-]
Failed to connect to the agent running on node d7c322f0-0354-4008-92b4-
↳ f49fb2201001
for invoking command clean.get_clean_steps. Error:
HTTPSConnectionPool(host='192.168.123.126', port=9999): Max retries exceeded.
↳ with url:
/v1/commands/?wait=true&agent_token=<token> (Caused by
SSLError(SSLError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify
↳ failed (_ssl.c:897)'),)):
requests.exceptions.SSLError: HTTPSConnectionPool(host='192.168.123.126',
↳ port=9999):
Max retries exceeded with url: /v1/commands/?wait=true&agent_token=<token>
(Caused by SSLError(SSLError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate
↳ verify failed (_ssl.c:897)'),))
```

The cause of the issue is that the Bare Metal service cannot access the ramdisk with the TLS certificate provided by the ramdisk on first heartbeat. You can inspect the stored certificate in `/var/lib/ironic/certificates/<node>.crt`.

You can try connecting to the ramdisk using the IP address in the log message:

```
curl -vL https://<IP address>:9999/v1/commands \  
--cacert /var/lib/ironic/certificates/<node UUID>.cert
```

You can get the detailed information about the certificate using openSSL:

```
openssl x509 -text -noout -in /var/lib/ironic/certificates/<node UUID>.cert
```

Clock skew

One possible source of the problem is a discrepancy between the hardware clock on the node and the time on the machine with the Bare Metal service. It can be detected by comparing the `Not Before` field in the `openssl` output with the timestamp of a log message.

The recommended solution is to enable the NTP support in `ironic-python-agent` by passing the `ipa-ntp-server` argument with an address of an NTP server reachable by the node.

If it is not possible, you need to ensure the correct hardware time on the machine. Keep in mind a potential issue with timezones: an ability to store timezone in hardware is pretty recent and may not be available. Since `ironic-python-agent` is likely operating in UTC, the hardware clock should also be set in UTC.

Note: Microsoft Windows uses local time by default, so a machine that has previously run Windows will likely have wrong time.

I changed `ironic.conf`, and now I cant edit my nodes.

Whenever a node is created in `ironic`, default interfaces are identified as part of driver composition. This maybe sourced from explicit default values which have been set in `ironic.conf` or by the interface order for the enabled interfaces list. The result of this is that the `ironic-conductor` cannot spawn a task using the composed driver, as a portion of the driver is no longer enabled. This makes it difficult to edit or update the node if the settings have been changed.

For example, with networking interfaces, if you have `default_network_interface=neutron` or `enabled_network_interfaces=neutron,flat` in your `ironic.conf`, nodes would have been created with the `neutron` network interface.

This is because `default_network_interface` overrides the setting for new nodes, and that setting is **saved** to the database nodes table.

Similarly, the order of `enabled_network_interfaces` takes priority, and the first entry in the list is generally set to the default for the node upon creation, and that record is **saved** to the database nodes table.

The only case where driver composition does *not* calculate a default is if an explicit value is provided upon the creation of the node.

Example failure

A node in this state, when the `network_interface` was saved as `neutron`, yet the `neutron` interface is no longer enabled will fail basic state transition requests.:

```
$ baremetal node manage 7164efca-37ab-1213-1112-b731cf795a5a Could not find the following interface in the ironic.hardware.interfaces.network entrypoint: neutron. Valid interfaces are [flat]. (HTTP 400)
```

How to fix this?

Revert the changes you made to `ironic.conf`.

This applies to any changes to any `default_*_interface` options or the order of interfaces in the for the `enabled_*_interfaces` options.

Once the conductor has been restarted with the updated configuration, you should now be able to update the interface using the `baremetal node set` command. In this example we use the `network_interface` as this is most commonly where it is encountered.:

```
$ baremetal node set $NAME_OR_UUID network-interface flat
```

Note: There are additional paths one can take to remedy this sort of issue, however we encourage operators to be mindful of operational consistency when making major configuration changes.

Once you have updated the saved interfaces, you should be able to safely return the `ironic.conf` configuration change in changing what interfaces are enabled by the conductor.

Im getting Out of Memory errors

This issue, also known as the the OOMKiller got my conductor case, is where your OS system memory reaches a point where the operating system engages measures to shed active memory consumption in order to prevent a complete failure of the machine. Unfortunately this can cause unpredictable behavior.

How did I get here?

One of the major consumers of memory in a host running an ironic-conductor is transformation of disk images using the `qemu-img` tool. This tool, because the disk images it works with are both compressed and out of linear block order, requires a considerable amount of memory to efficiently re-assemble and write-out a disk to a device, or to simply convert the format such as to a raw image.

By default, ironics configuration limits this conversion to 1 GB of RAM for the process, but each conversion does cause additional buffer memory to be used, which increases overall system memory pressure. Generally memory pressure alone from buffers will not cause an out of memory condition, but the multiple conversions or deployments running at the same time CAN cause extreme memory pressure and risk the system running out of memory.

How do I resolve this?

This can be addressed a few different ways:

- Use raw images, however these images can be substantially larger and require more data to be transmitted over the wire.
- Add more physical memory.
- Add swap space.
- Reduce concurrency, possibly via another conductor or changing the `nova-compute.conf` `max_concurrent_builds` parameter.
- Or finally, adjust the `[DEFAULT]minimum_required_memory` parameter in your `ironic.conf` file. The default should be considered a default of last resort and you may need to reserve additional memory. You may also wish to adjust the `[DEFAULT]minimum_memory_wait_retries` and `[DEFAULT]minimum_memory_wait_time` parameters.

4.2.18 Power Sync with the Compute Service

Baremetal Power Sync

Each Baremetal conductor process runs a periodic task which synchronizes the power state of the nodes between its database and the actual hardware. If the value of the `conductor.force_power_state_during_sync` option is set to `true` the power state in the database will be forced on the hardware and if it is set to `false` the hardware state will be forced on the database. If this periodic task is enabled, it runs at an interval defined by the `conductor.sync_power_state_interval` config option for those nodes which are not in maintenance.

Compute-Baremetal Power Sync

Each `nova-compute` process in the Compute service runs a periodic task which synchronizes the power state of servers between its database and the compute driver. If enabled, it runs at an interval defined by the `sync_power_state_interval` config option on the `nova-compute` process. In case of the compute driver being baremetal driver, this sync will happen between the databases of the compute and baremetal services. Since the sync happens on the `nova-compute` process, the state in the compute database will be forced on the baremetal database in case of inconsistencies. Hence a node which was put down using the compute service API cannot be brought up through the baremetal service API since the power sync task will regard the compute services knowledge of the power state as the source of truth. In order to get around this disadvantage of the compute-baremetal power sync, baremetal service does power state change callbacks to the compute service using external events.

Power State Change Callbacks to the Compute Service

Whenever the Baremetal service changes the power state of a node, it can issue a notification to the Compute service. The Compute service will consume this notification and update the power state of the instance in its database. By conveying all the power state changes to the compute service, the baremetal service becomes the source of truth thus preventing the compute service from forcing wrong power states on the physical instance during the compute-baremetal power sync. It also adds the possibility of bringing up/down a physical instance through the baremetal service API even if it was put down/up through the compute service API.

This change requires the *nova* section and the necessary authentication options like the *nova.auth_url* to be defined in the configuration file of the baremetal service. If it is not configured the baremetal service will not be able to send notifications to the compute service and it will fall back to the behaviour of the compute service forcing power states on the baremetal service during the power sync. See *nova* group for more details on the available config options.

In case of baremetal stand alone deployments where there is no compute service running, the *nova.send_power_notifications* config option should be set to `False` to disable power state change callbacks to the compute service.

Note: The baremetal service sends notifications to the compute service only if the target power state is `power on` or `power off`. Other error and `None` states will be ignored. In situations where the power state change is originally coming from the compute service, the notification will still be sent by the baremetal service and it will be a no-op on the compute service side with a debug log stating the node is already powering on/off.

Note: Although an exclusive lock is used when sending notifications to the compute service, there can still be a race condition if the compute-baremetal power sync happens to happen a nano-second before the power state change event is received from the baremetal service in which case the power state from compute services database will be forced on the node.

4.2.19 Node Multi-Tenancy

This guide explains the steps needed to enable node multi-tenancy. This feature enables non-admins to perform API actions on nodes, limited by policy configuration. The Bare Metal service supports two kinds of non-admin users:

- **Owner:** owns specific nodes and performs administrative actions on them
- **Lessee:** receives temporary and limited access to a node

Setting the Owner and Lessee

Non-administrative access to a node is controlled through a nodes owner or lessee attribute:

```
baremetal node set --owner 080925ee2f464a2c9dce91ee6ea354e2 node-7
baremetal node set --lessee 2a210e5ff114c8f2b6e994218f51a904 node-10
```

Configuring the Bare Metal Service Policy

By default, the Bare Metal service policy is configured so that a node owner or lessee has no access to any node APIs. However, the policy *policy file* contains rules that can be used to enable node API access:

```
# Owner of node
#"is_node_owner": "project_id:%(node.owner)s"

# Lessee of node
#"is_node_lessee": "project_id:%(node.lessee)s"
```

An administrator can then modify the policy file to expose individual node APIs as follows:

```
# Change Node provision status
# PUT /nodes/{node_ident}/states/provision
#"baremetal:node:set_provision_state": "rule:is_admin"
"baremetal:node:set_provision_state": "rule:is_admin or rule:is_node_owner or_
↳rule:is_node_lessee"

# Update Node records
# PATCH /nodes/{node_ident}
#"baremetal:node:update": "rule:is_admin or rule:is_node_owner"
```

In addition, it is safe to expose the `baremetal:node:list` rule, as the node list function now filters non-admins by owner and lessee:

```
# Retrieve multiple Node records, filtered by owner
# GET /nodes
# GET /nodes/detail
#"baremetal:node:list": "rule:baremetal:node:get"
"baremetal:node:list": ""
```

Note that `baremetal:node:list_all` permits users to see all nodes regardless of owner/lessee, so it should remain restricted to admins.

Ports

Port APIs can be similarly exposed to node owners and lessees:

```
# Retrieve Port records
# GET /ports/{port_id}
# GET /nodes/{node_ident}/ports
# GET /nodes/{node_ident}/ports/detail
# GET /portgroups/{portgroup_ident}/ports
# GET /portgroups/{portgroup_ident}/ports/detail
#"baremetal:port:get": "rule:is_admin or rule:is_observer"
"baremetal:port:get": "rule:is_admin or rule:is_observer or rule:is_node_
↳owner or rule:is_node_lessee"

# Retrieve multiple Port records, filtered by owner
# GET /ports
# GET /ports/detail
#"baremetal:port:list": "rule:baremetal:port:get"
"baremetal:port:list": ""
```

Allocations

Allocations respect node tenancy as well. A restricted allocation creates an allocation tied to a project, and that can only match nodes where that project is the owner or lessee. Here is a sample set of allocation policy rules that allow non-admins to use allocations effectively:

```
# Retrieve Allocation records
# GET /allocations/{allocation_id}
# GET /nodes/{node_ident}/allocation
#"baremetal:allocation:get": "rule:is_admin or rule:is_observer"
"baremetal:allocation:get": "rule:is_admin or rule:is_observer or rule:is_
↳allocation_owner"

# Retrieve multiple Allocation records, filtered by owner
# GET /allocations
#"baremetal:allocation:list": "rule:baremetal:allocation:get"
"baremetal:allocation:list": ""

# Retrieve multiple Allocation records
# GET /allocations
#"baremetal:allocation:list_all": "rule:baremetal:allocation:get"

# Create Allocation records
# POST /allocations
#"baremetal:allocation:create": "rule:is_admin"

# Create Allocation records that are restricted to an owner
# POST /allocations
#"baremetal:allocation:create_restricted": "rule:baremetal:allocation:create"
```

(continues on next page)

(continued from previous page)

```

"baremetal:allocation:create_restricted": ""

# Delete Allocation records
# DELETE /allocations/{allocation_id}
# DELETE /nodes/{node_id}/allocation
#"baremetal:allocation:delete": "rule:is_admin"
"baremetal:allocation:delete": "rule:is_admin or rule:is_allocation_owner"

# Change name and extra fields of an allocation
# PATCH /allocations/{allocation_id}
#"baremetal:allocation:update": "rule:is_admin"
"baremetal:allocation:update": "rule:is_admin or rule:is_allocation_owner"

```

Deployment and Metalsmith

Provisioning a node requires a specific set of APIs to be made available. The following policy specifications are enough to allow a node owner to use [Metalsmith](#) to deploy upon a node:

```

"baremetal:node:get": "rule:is_admin or rule:is_observer or rule:is_node_owner
↪"
"baremetal:node:list": ""
"baremetal:node:update_extra": "rule:is_admin or rule:is_node_owner"
"baremetal:node:update_instance_info": "rule:is_admin or rule:is_node_owner"
"baremetal:node:validate": "rule:is_admin or rule:is_node_owner"
"baremetal:node:set_provision_state": "rule:is_admin or rule:is_node_owner"
"baremetal:node:vif:list": "rule:is_admin or rule:is_node_owner"
"baremetal:node:vif:attach": "rule:is_admin or rule:is_node_owner"
"baremetal:node:vif:detach": "rule:is_admin or rule:is_node_owner"
"baremetal:allocation:get": "rule:is_admin or rule:is_observer or rule:is_
↪allocation_owner"
"baremetal:allocation:list": ""
"baremetal:allocation:create_restricted": ""
"baremetal:allocation:delete": "rule:is_admin or rule:is_allocation_owner"
"baremetal:allocation:update": "rule:is_admin or rule:is_allocation_owner"

```

4.2.20 Fast-Track Deployment

Fast track is a mode of operation where the Bare Metal service keeps a machine powered on with the agent running between provisioning operations. It is first booted during in-band inspection or cleaning (whatever happens first) and is only shut down before rebooting into the final instance. Depending on the configuration, this mode can save several reboots and is particularly useful for scenarios where nodes are enrolled, prepared and provisioned within a short period of time.

Warning: Fast track deployment targets standalone use cases and is only tested with the noop networking. The case where inspection, cleaning and provisioning networks are different is not supported.

Enabling

Fast track is off by default and should be enabled in the configuration:

```
[deploy]
fast_track = true
```

Inspection

If using *In-band inspection*, you need to tell ironic-inspector not to power off nodes afterwards. Depending on the inspection mode (managed or unmanaged), you need to configure two places. In `ironic.conf`:

```
[inspector]
power_off = false
```

And in `inspector.conf`:

```
[processing]
power_off = false
```

Finally, you need to update the `inspection PXE configuration` to include the `ipa-api-url` kernel parameter, pointing at the **ironic** endpoint, in addition to the existing `ipa-inspection-callback-url`.

4.2.21 Booting a Ramdisk or an ISO

Ironic supports booting a user provided ramdisk or an ISO image (starting with the Victoria release) instead of deploying a node. Most commonly this is performed when an instance is booted via PXE, iPXE or Virtual Media, with the only local storage contents being those in memory. It is supported by `pxe`, `ipxe`, `redfish-virtual-media` and `ilo-virtual-media` boot interfaces.

Configuration

Ramdisk/ISO boot requires using the `ramdisk` deploy interface. As with most non-default interfaces, it must be enabled and set for a node to be utilized:

```
[DEFAULT]
...
enabled_deploy_interfaces = direct,ramdisk
...
```

Once enabled and the conductor(s) have been restarted, the interface can be set upon creation of a new node:

```
baremetal node create --driver ipmi \
  --deploy-interface ramdisk \
  --boot-interface ipxe
```

or update an existing node:

```
baremetal node set <NODE> --deploy-interface ramdisk
```

You can also use it with *redfish virtual media* instead of iPXE.

Creating a ramdisk

A ramdisk can be created using the `ironic-ramdisk-base` element from `ironic-python-agent-builder`, e.g. with Debian:

```
export ELEMENTS_PATH=/opt/stack/ironic-python-agent-builder/dib
disk-image-create -o /output/ramdisk \
  debian-minimal ironic-ramdisk-base openssh-server dhcp-all-interfaces
```

You should consider using the following elements:

- `openssh-server` to install the SSH server since its not provided by default by some minimal images.
- `devuser` or `dynamic-login` to provide SSH access.
- `dhcp-all-interfaces` or `simple-init` to configure networking.

The resulting files (`/output/ramdisk.kernel` and `/output/ramdisk.initramfs` in this case) can then be used when *Booting a ramdisk*.

Booting a ramdisk

Pass the kernel and ramdisk as normally, also providing the ramdisk as an image source, for example,

```
baremetal node set <NODE> \
  --instance-info kernel=http://path/to/ramdisk.kernel \
  --instance-info ramdisk=http://path/to/ramdisk.initramfs \
  --instance-info image_source=http://path/to/ramdisk.initramfs
baremetal node deploy <NODE>
```

Note: The requirement to pass `image_source` is artificial and will be fixed in a future version of the Bare Metal service.

Booting an ISO

The ramdisk deploy interface can also be used to boot an ISO image. For example,

```
baremetal node set <NODE> \
  --instance-info boot_iso=http://path/to/boot.iso
baremetal node deploy <NODE>
```

Warning: This feature, when utilized with the `ipxe boot_interface`, will only allow a kernel and ramdisk to be booted from the supplied ISO file. Any additional contents, such as additional ramdisk

contents or installer package files will be unavailable after the boot of the Operating System. Operators wishing to leverage this functionality for actions such as OS installation should explore use of the standard `ramdisk_deploy_interface` along with the `instance_info/kernel_append_params` setting to pass arbitrary settings such as a mirror URL for the initial ramdisk to load data from. This is a limitation of iPXE and the overall boot process of the operating system where memory allocated by iPXE is released.

Limitations

The intended use case is for advanced scientific and ephemeral workloads where the step of writing an image to the local storage is not required or desired. As such, this interface does come with several caveats:

- Configuration drives are not supported with network boot, only with Redfish virtual media.
- Disk image contents are not written to the bare metal node.
- Users and Operators who intend to leverage this interface should expect to leverage a metadata service, custom ramdisk images, or the `instance_info/ramdisk_kernel_arguments` parameter to add options to the kernel boot command line.
- When using PXE/iPXE boot, bare metal nodes must continue to have network access to PXE and iPXE network resources. This is contrary to most tenant networking enabled configurations where this access is restricted to the provisioning and cleaning networks
- As with all deployment interfaces, automatic cleaning of the node will still occur with the contents of any local storage being wiped between deployments.

4.2.22 Drivers, Hardware Types and Hardware Interfaces

4.2.23 Advanced Topics

Ceph Object Gateway support

Overview

Ceph project is a powerful distributed storage system. It contains object store and provides a RADOS Gateway Swift API which is compatible with OpenStack Swift API.

Ironic added support for RADOS Gateway temporary URL in the Mitaka release.

Configure Ironic and Glance with RADOS Gateway

1. Install Ceph storage with RADOS Gateway. See [Ceph documentation](#).
2. Configure RADOS Gateway to use keystone for authentication. See [Integrating with OpenStack Keystone](#)
3. Register RADOS Gateway endpoint in the keystone catalog, with the same format swift uses, as the `object-store` service. URL example:

```
http://rados.example.com:8080/swift/v1/AUTH_$(project_id)s.
```


In the ceph configuration, make sure radosgw is configured with the following value:

```
rgw swift account in url = True
```

4. Configure Glance API service for RADOS Swift API as backend. Edit the configuration file for the Glance API service (is typically located at `/etc/glance/glance-api.conf`):

```
[glance_store]

stores = file, http, swift
default_store = swift
default_swift_reference=ref1
swift_store_config_file=/etc/glance/glance-swift-creds.conf
swift_store_container = glance
swift_store_create_container_on_put = True
```

In the file referenced in `swift_store_config_file` option, add the following:

```
[ref1]
user = <service project>:<service user name>
key = <service user password>
user_domain_id = default
project_domain_id = default
auth_version = 3
auth_address = http://keystone.example.com/identity
```

Values for `user` and `key` options correspond to keystone credentials for RADOS Gateway service user.

Note: RADOS Gateway uses FastCGI protocol for interacting with HTTP server. Read your HTTP server documentation if you want to enable HTTPS support.

5. Restart Glance API service and upload all needed images.
6. If you're using custom container name in RADOS, change Ironic configuration file on the conductor host(s) as follows:

```
[glance]

swift_container = glance
```

7. Restart Ironic conductor service(s).

Building images for Windows

We can use `New-WindowsOnlineImage` in `windows-openstack-imaging-tools` tool as an option to create Windows images (whole disk images) corresponding boot modes which will support for Windows NIC Teaming. And allow the utilization of link aggregation when the instance is spawned on hardware servers (Bare metals).

Requirements:

- A Microsoft Windows Server Operating System along with Hyper-V virtualization enabled, PowerShell version ≥ 4 supported, Windows Assessment and Deployment Kit, in short Windows ADK.
- The windows Server compatible drivers.
- Working git environment.

Preparation:

- Download a Windows Server 2012R2/ 2016 installation ISO.
- Install Windows Server 2012R2/ 2016 OS on workstation PC along with following feature:
 - Enable Hyper-V virtualization.
 - Install PowerShell 4.0.
 - Install Git environment & import git proxy (if have).
 - Create new Path in Microsoft Windows Server Operating System which support for submodule update via `git submodule update init` command:

```
- Variable name: Path
- Variable value: C:\Windows\System32\WindowsPowerShell\v1.0\;C:\
↪Program Files\Git\bin
```

- Rename virtual switch name in Windows Server 2012R2/ 2016 in Virtual Switch Manager into *external*.

Implementation:

- Step 1: Create folders: `C:\<folder_name_1>` where output images will be located, `C:\<folder_name_2>` where you need to place the necessary hardware drivers.
- Step 2: Copy and extract necessary hardware drivers in `C:\<folder_name_2>`.
- Step 3: Insert or burn Windows Server 2016 ISO to `D:\`.
- Step 4: Download `windows-openstack-imaging-tools` tools.

```
git clone https://github.com/cloudbase/windows-openstack-imaging-tools.git
```

- Step 5: Create & running script `create-windows-cloud-image.ps1`:

```
git submodule update --init
Import-Module WinImageBuilder.psml
$windowsImagePath = "C:\<folder_name_1>\<output_file_name>.qcow2"
$virtIOISOPath = "C:\<folder_name_1>\virtio.iso"
$virtIODownloadLink = "https://fedorapeople.org/groups/virt/virtio-win/
↪direct-downloads/archive-virtio/virtio-win-0.1.133-2/virtio-win.iso"
(New-Object System.Net.WebClient).DownloadFile($virtIODownloadLink,
↪$virtIOISOPath)
```

(continues on next page)

(continued from previous page)

```

$wimFilePath = "D:\sources\install.wim"
$extraDriversPath = "C:\<folder_name_2>\\"
$image = (Get-WimFileImagesInfo -WimFilePath $wimFilePath)[1]
$switchName = 'external'
New-WindowsOnlineImage -WimFilePath $wimFilePath
  -ImageName $image.ImageName ` -WindowsImagePath $windowsImagePath -Type
↪ 'KVM' -ExtraFeatures @() `
  -SizeBytes 20GB -CpuCores 2 -Memory 2GB -SwitchName $switchName ` -
↪ ProductKey $productKey -DiskLayout 'BIOS' `
  -ExtraDriversPath $extraDriversPath ` -InstallUpdates:$false -
↪ AdministratorPassword 'Pa$$w0rd' `
  -PurgeUpdates:$true -DisableSwap:$true

```

After executing this command you will get two output files, first one being C:<folder_name_1><output_file_name>.qcow2, which is the resulting windows whole disk image and C:<folder_name_1>virtio.iso, which is virtio iso contains all the synthetic drivers for the KVM hypervisor.

See [example_windows_images](#) for more details and examples.

Note: We can change SizeBytes, CpuCores and Memory depending on requirements.

Emitting Software Metrics

Beginning with the Newton (6.1.0) release, the ironic services support emitting internal performance data to `statsd`. This allows operators to graph and understand performance bottlenecks in their system.

This guide assumes you have a `statsd` server setup. For information on using and configuring `statsd`, please see the [statsd](#) README and documentation.

These performance measurements, herein referred to as metrics, can be emitted from the Bare Metal service, including `ironic-api`, `ironic-conductor`, and `ironic-python-agent`. By default, none of the services will emit metrics.

Configuring the Bare Metal Service to Enable Metrics

Enabling metrics in `ironic-api` and `ironic-conductor`

The `ironic-api` and `ironic-conductor` services can be configured to emit metrics to `statsd` by adding the following to the ironic configuration file, usually located at `/etc/ironic/ironic.conf`:

```

[metrics]
backend = statsd

```

If a `statsd` daemon is installed and configured on every host running an ironic service, listening on the default UDP port (8125), no further configuration is needed. If you are using a remote `statsd` server, you must also supply connection information in the ironic configuration file:

```
[metrics_statsd]
# Point this at your environments' statsd host
statsd_host = 192.0.2.1
statsd_port = 8125
```

Enabling metrics in ironic-python-agent

The ironic-python-agent process receives its configuration in the response from the initial lookup request to the ironic-api service. This means to configure ironic-python-agent to emit metrics, you must enable the agent metrics backend in your ironic configuration file on all ironic-conductor hosts:

```
[metrics]
agent_backend = statsd
```

In order to reliably emit metrics from the ironic-python-agent, you must provide a statsd server that is reachable from both the configured provisioning and cleaning networks. The agent statsd connection information is configured in the ironic configuration file as well:

```
[metrics_statsd]
# Point this at a statsd host reachable from the provisioning and cleaning_
↔nets
agent_statsd_host = 198.51.100.2
agent_statsd_port = 8125
```

Types of Metrics Emitted

The Bare Metal service emits timing metrics for every API method, as well as for most driver methods. These metrics measure how long a given method takes to execute.

A deployer with metrics enabled should expect between 100 and 500 distinctly named data points to be emitted from the Bare Metal service. This will increase if the `metrics.preserve_host` option is set to true or if multiple drivers are used in the Bare Metal deployment. This estimate may be used to determine if a deployer needs to scale their metrics backend to handle the additional load before enabling metrics. To see which metrics have changed names or have been removed between releases, refer to the [ironic release notes](#).

Note: With the default statsd configuration, each timing metric may create additional metrics due to how statsd handles timing metrics. For more information, see [statsd documentation on metric types](#).

The ironic-python-agent ramdisk emits timing metrics for every API method.

Deployers who use custom HardwareManagers can emit custom metrics for their hardware. For more information on custom HardwareManagers, and emitting metrics from them, please see the [ironic-python-agent documentation](#).

Adding New Metrics

If you're a developer, and would like to add additional metrics to ironic, please see the [ironic-lib developer documentation](#) for details on how to use the metrics library. A release note should also be created each time a metric is changed or removed to alert deployers of the change.

API Audit Logging

Audit middleware supports delivery of CADF audit events via Oslo messaging notifier capability. Based on *notification_driver* configuration, audit events can be routed to messaging infrastructure (*notification_driver = messagingv2*) or can be routed to a log file (*[oslo_messaging_notifications]/driver = log*).

Audit middleware creates two events per REST API interaction. First event has information extracted from request data and the second one has request outcome (response).

Enabling API Audit Logging

Audit middleware is available as part of *keystonemiddleware* (≥ 1.6) library. For information regarding how audit middleware functions refer [here](#).

Auditing can be enabled for the Bare Metal service by making the following changes to `/etc/ironic/ironic.conf`.

1. To enable audit logging of API requests:

```
[audit]
...
enabled=true
```

2. To customize auditing API requests, the audit middleware requires the `audit_map_file` setting to be defined. Update the value of configuration setting `audit_map_file` to set its location. Audit map file configuration options for the Bare Metal service are included in the `etc/ironic/ironic_api_audit_map.conf.sample` file. To understand CADF format specified in `ironic_api_audit_map.conf` file refer to [CADF Format](#):

```
[audit]
...
audit_map_file=/etc/ironic/api_audit_map.conf
```

3. Comma separated list of Ironic REST API HTTP methods to be ignored during audit. It is used only when API audit is enabled. For example:

```
[audit]
...
ignore_req_list=GET,POST
```

Sample Audit Event

Following is the sample of audit event for ironic node list request.

```
{
  "event_type": "audit.http.request",
  "timestamp": "2016-06-15 06:04:30.904397",
  "payload": {
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    "eventTime": "2016-06-15T06:04:30.903071+0000",
    "target": {
      "id": "ironic",
      "typeURI": "unknown",
      "addresses": [
        {
          "url": "http://{ironic_admin_host}:6385",
          "name": "admin"
        },
        {
          "url": "http://{ironic_internal_host}:6385",
          "name": "private"
        },
        {
          "url": "http://{ironic_public_host}:6385",
          "name": "public"
        }
      ],
      "name": "ironic"
    },
    "observer": {
      "id": "target"
    },
    "tags": [
      "correlation_id?value=685f1abb-620e-5d5d-b74a-b4135fb32373"
    ],
    "eventType": "activity",
    "initiator": {
      "typeURI": "service/security/account/user",
      "name": "admin",
      "credential": {
        "token": "****",
        "identity_status": "Confirmed"
      },
      "host": {
        "agent": "python-ironicclient",
        "address": "10.1.200.129"
      },
      "project_id": "d8f52dd7d9e1475dbbf3ba47a4a83313",
      "id": "8c1a948bad3948929aa5d5b50627a174"
    },
  },
}
```

(continues on next page)

(continued from previous page)

```
"action": "read",
"outcome": "pending",
"id": "061b7aa7-5879-5225-a331-c002cf23cb6c",
"requestPath": "/v1/nodes/?associated=True"
},
"priority": "INFO",
"publisher_id": "ironic-api",
"message_id": "2f61ebaa-2d3e-4023-afba-f9fca6f21fc2"
}
```

Bare Metal Service state report (via Guru Meditation Reports)

The Bare Metal service contains a mechanism whereby developers and system administrators can generate a report about the state of running Bare Metal executables (`ironic-api` and `ironic-conductor`). This report is called a Guru Meditation Report (GMR for short). GMR provides useful debugging information that can be used to obtain an accurate view on the current live state of the system. For example, what threads are running, what configuration parameters are in effect, and more. The eventlet backdoor facility provides an interactive shell interface for any eventlet based process, allowing an administrator to telnet to a pre-defined port and execute a variety of commands.

Configuration

The GMR feature is optional and requires the `oslo.reports` package to be installed. For example, using `pip`:

```
pip install 'oslo.reports>=1.18.0'
```

Generating a GMR

A *GMR* can be generated by sending the `USR2` signal to any Bare Metal process that supports it. The *GMR* will then be output to `stderr` for that particular process. For example:

Suppose that `ironic-api` has process ID 6385, and was run with `2>/var/log/ironic/ironic-api-err.log`. Then, sending the `USR` signal:

```
kill -USR2 6385
```

will trigger the Guru Meditation report to be printed to `/var/log/ironic/ironic-api-err.log`.

Structure of a GMR

The *GMR* consists of the following sections:

Package Shows information about the package to which this process belongs, including version information.

Threads Shows stack traces and thread IDs for each of the threads within this process.

Green Threads Shows stack traces for each of the green threads within this process (green threads don't have thread IDs).

Configuration Lists all the configuration options currently accessible via the CONF object for the current process.

Agent Token

Purpose

The concept of agent tokens is to provide a mechanism by which the relationship between an operating deployment of the Bare Metal Service and an instance of the `ironic-python-agent` is verified. In a sense, this token can be viewed as a session identifier or authentication token.

Warning: This functionality does not remove the risk of a man-in-the-middle attack that could occur from connection intercept or when TLS is not used for all communication.

This becomes useful in the case of deploying an edge node where intermediate networks are not trustworthy.

How it works

These tokens are provided in one of two ways to the running agent.

1. A pre-generated token which is embedded into virtual media ISOs.
2. A one-time generated token that are provided upon the first lookup of the node.

In both cases, the tokens are a randomly generated using the Python `secrets` library. As of mid-2020, the default length is 43 characters.

Once the token has been provided, the token cannot be retrieved or accessed. It remains available to the conductors, and is stored in memory of the `ironic-python-agent`.

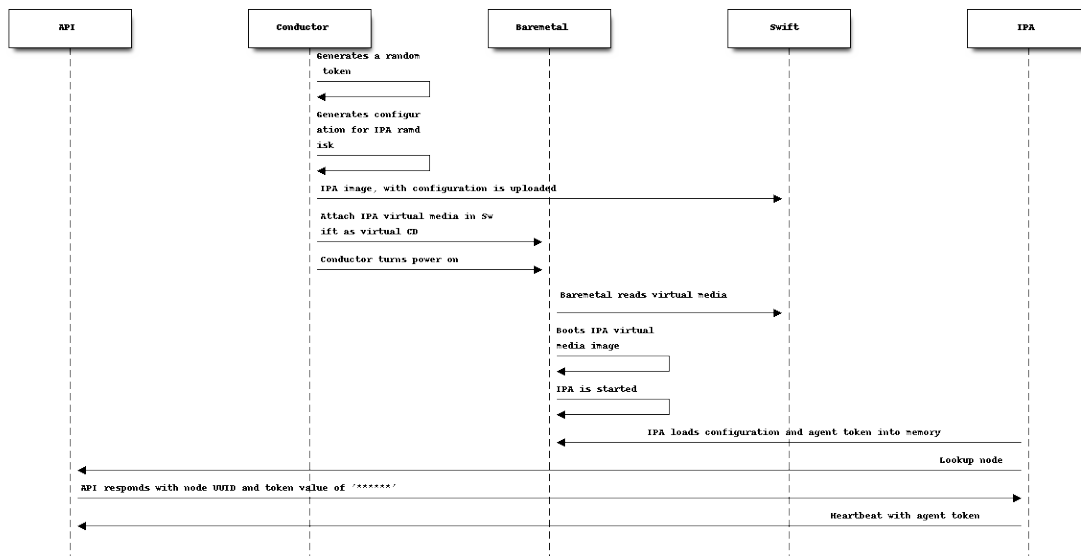
Note: In the case of the token being embedded with virtual media, it is read from a configuration file with-in the image. Ideally this should be paired with Swift temporary URLs.

With the token is available in memory in the agent, the token is embedded with `heartbeat` operations to the `ironic` API endpoint. This enables the API to authenticate the heartbeat request, and refuse heartbeat requests from the `ironic-python-agent`. As of the Victoria release, use of Agent Token is required for all agents and the previously available setting to force this functionality to be mandatory, `[DEFAULT]require_agent_token` no longer has any effect.

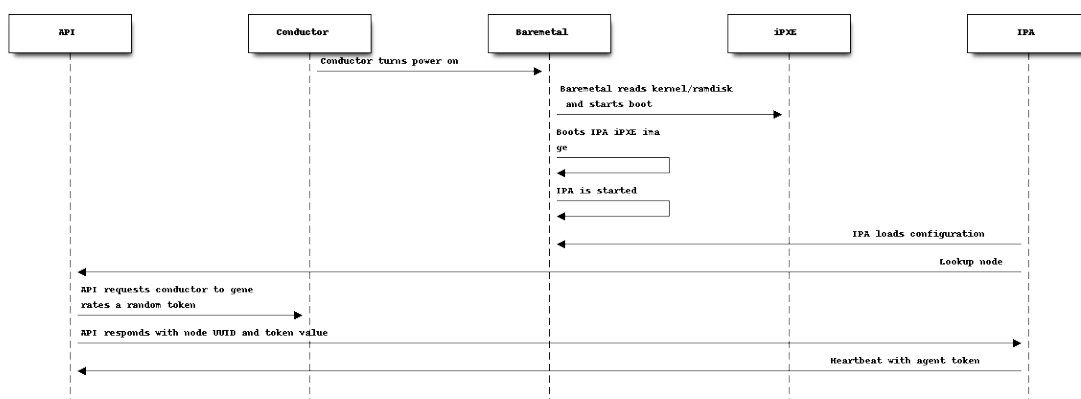
Warning: If the Bare Metal Service is updated, and the version of `ironic-python-agent` should be updated to enable this feature.

In addition to heartbeats being verified, commands from the `ironic-conductor` service to the `ironic-python-agent` also include the token, allowing the agent to authenticate the caller.

With Virtual Media



With PXE/IPXE/etc.



Agent Configuration

An additional setting which may be leveraged with the `ironic-python-agent` is a `agent_token_required` setting. Under normal circumstances, this setting can be asserted via the configuration supplied from the Bare Metal service deployment upon the `lookup` action, but can be asserted via the embedded configuration for the agent in the ramdisk. This setting is also available via kernel command line as `ipa-agent-token-required`.

Deploying without BMC Credentials

The Bare Metal service usually requires BMC credentials for all provisioning operations. Starting with the Victoria release series there is limited support for inspection, cleaning and deployments without the credentials.

Warning: This feature is experimental and only works in a limited scenario. When using it, you have to be prepared to provide BMC credentials in case of a failure or any non-supported actions.

How it works

The expected workflow is as follows:

1. The node is discovered by manually powering it on and gets the *manual-management* hardware type and *agent* power interface.
If discovery is not used, a node can be enrolled through the API and then powered on manually.
2. The operator moves the node to *manageable*. It works because the *agent* power only requires to be able to connect to the agent.
3. The operator moves the node to *available*. Cleaning happens normally via the already running agent. If reboot is needed, it is done by telling the agent to reboot the node in-band.
4. A user deploys the node. Deployment happens normally via the already running agent.
5. In the end of the deployment, the node is rebooted via the `reboot` command instead of `power off+on`.

Enabling

Fast-Track Deployment is a requirement for this feature to work. After enabling it, adds the *agent* power interface and the `manual-management` hardware type to the enabled list:

```
[DEFAULT]
enabled_hardware_types = manual-management
enabled_management_interfaces = noop
enabled_power_interfaces = agent

[deploy]
fast_track = true
```

As usual with the `noop` management, enable the networking boot fallback:

```
[pxe]
enable_netboot_fallback = true
```

If using discovery, configure discovery in `ironic-inspector` with the default driver set to `manual-management`.

Limitations

- Only the noop network interface is supported.
- Undeploy and rescue are not supported, you need to add BMC credentials first.
- If any errors happens in the process, recovery will likely require BMC credentials.
- Only rebooting is possible through the API, power on/off commands will fail.

Layer 3 or DHCP-less ramdisk booting

Bootting nodes via PXE, while universally supported, suffers from one disadvantage: it requires a direct L2 connectivity between the node and the control plane for DHCP. Using virtual media it is possible to avoid not only the unreliable TFTP protocol, but DHCP altogether.

When network data is provided for a node as explained below, the generated virtual media ISO will also serve as a `configdrive`, and the network data will be stored in the standard OpenStack location.

The `simple-init` element needs to be used when creating the deployment ramdisk. The `Glean` tool will look for a media labeled as `config-2`. If found, the network information from it will be read, and the nodes networking stack will be configured accordingly.

```
ironic-python-agent-builder -o /output/ramdisk \
    debian-minimal -e simple-init
```

Warning: Ramdisks based on distributions with NetworkManager require `Glean 1.19.0` or newer to work.

Note: If desired, some interfaces can still be configured to use DHCP.

Hardware type support

This feature is known to work with the following hardware types:

- *Redfish* with `redfish-virtual-media` boot
- *iLO* with `ilo-virtual-media` boot

Configuring network data

When the Bare Metal service is running within OpenStack, no additional configuration is required - the network configuration will be fetched from the Network service.

Alternatively, the user can build and pass network configuration in form of a `network_data` JSON to a node via the `network_data` field. Node-based configuration takes precedence over the configuration generated by the Network service and also works in standalone mode.

```
baremetal node set --network-data ~/network_data.json <node>
```

An example network data:

```
{
  "links": [
    {
      "id": "port-92750f6c-60a9-4897-9cd1-090c5f361e18",
      "type": "phy",
      "ethernet_mac_address": "52:54:00:d3:6a:71"
    }
  ],
  "networks": [
    {
      "id": "network0",
      "type": "ipv4",
      "link": "port-92750f6c-60a9-4897-9cd1-090c5f361e18",
      "ip_address": "192.168.122.42",
      "netmask": "255.255.255.0",
      "network_id": "network0",
      "routes": []
    }
  ],
  "services": []
}
```

Note: Some fields are redundant with the port information. Were looking into simplifying the format, but currently all these fields are mandatory.

You'll need the deployed image to support network data, e.g. by pre-installing `cloud-init` or `Glean` on it (most cloud images have the former). Then you can provide the network data when deploying, for example:

```
baremetal node deploy <node> \
  --config-drive "{\"network_data\": $(cat ~/network_data.json)}"
```

Some first-boot services, such as `Ignition`, don't support network data. You can provide their configuration as part of user data instead:

```
baremetal node deploy <node> \
  --config-drive "{\"user_data\": \"... ignition config ...\"}"
```

Deploying outside of the provisioning network

If you need to combine traditional deployments using a provisioning network with virtual media deployments over L3, you may need to provide an alternative IP address for the remote nodes to connect to:

```
[deploy]
http_url = <HTTP server URL internal to the provisioning network>
external_http_url = <HTTP server URL with a routable IP address>
```

You may also need to override the callback URL, which is normally fetched from the service catalog or configured in the [service_catalog] section:

```
[deploy]
external_callback_url = <Bare Metal API URL with a routable IP address>
```

Tuning Ironic

Memory Utilization

Memory utilization is a difficult thing to tune in Ironic as largely we may be asked by API consumers to perform work for which the underlying tools require large amounts of memory.

The biggest example of this is image conversion. Images not in a raw format need to be written out to disk for conversion (when requested) which requires the conversion process to generate an in-memory map to re-assemble the image contents into a coherent stream of data. This entire process also stresses the kernel buffers and cache.

This ultimately comes down to a trade-off of Memory versus Performance, similar to the trade-off of Performance versus Cost.

On a plus side, an idle Ironic deployment does not need much in the way of memory. On the down side, a highly bursty environment where a large number of concurrent deployments may be requested should consider two aspects:

- How is the ironic-api service/process set up? Will more processes be launched automatically?
- Are images prioritized for storage size on disk? Or are they compressed and require format conversion?

API

Ironic's API should have a fairly stable memory footprint with activity, however depending on how the webserver is running the API, additional processes can be launched.

Under normal conditions, as of Ironic 15.1, the `ironic-api` service/process consumes approximately 270MB of memory per worker. Depending on how the process is being launched, the number of workers and maximum request threads per worker may differ. Naturally there are configuration and performance trade-offs.

- Directly as a native python process, i.e. execute `ironic-api` processes. Each single worker allows for multiple requests to be handled and threaded at the same time which can allow high levels of

request concurrency. As of the Victoria cycle, a direct invocation of the `ironic-api` program will only launch a maximum of four workers.

- Launched via a wrapper such as Apache+uWSGI may allow for multiple distinct worker processes, but these workers typically limit the number of request processing threads that are permitted to execute. This means requests can stack up in the front-end webserver and be released to the `ironic-api` as prior requests complete. In environments with long running synchronous calls, such as use of the vendor passthru interface, this can be very problematic.

When the webserver is launched by the API process directly, the default is based upon the number of CPU sockets in your machine.

When launching using `uwsgi`, this will entirely vary upon your configuration, but balancing workers/threads based upon your load and needs is highly advisable. Each worker process is unique and consumes far more memory than a comparable number of worker threads. At the same time, the scheduler will focus on worker processes as the threads are greenthreads.

Note: Host operating systems featuring in-memory de-duplication should see an improvement in the overall memory footprint with multiple processes, but this is not something the development team has measured and will vary based upon multiple factors.

One important item to note: each Ironic API service/process *does* keep a copy of the hash ring as generated from the database *in-memory*. This is done to help allocate load across a cluster in-line with how individual nodes and their responsible conductors are allocated across the cluster. In other words, your amount of memory **WILL** increase corresponding to the number of nodes managed by each ironic conductor. It is important to understand that features such as `conductor groups` means that only matching portions of nodes will be considered for the hash ring if needed.

Conductor

A conductor process will launch a number of other processes, as required, in order to complete the requested work. Ultimately this means it can quickly consume large amounts of memory because it was asked to complete a substantial amount of work all at once.

The `ironic-conductor` from ironic 15.1 consumes by default about 340MB of RAM in an idle configuration. This process, by default, operates as a single process. Additional processes can be launched, but they must have unique resolvable hostnames and addresses for JSON-RPC or use a central oslo.messaging supported message bus in order for Webserver API to Conductor API communication to be functional.

Typically, the most memory intensive operation that can be triggered is a image conversion for deployment, which is limited to 1GB of RAM per conversion process.

Most deployments, by default, do have a concurrency limit depending on their Compute (See `nova.conf` setting `max_concurrent_builds`) configuration. However, this is only per `nova-compute` worker, so naturally this concurrency will scale with additional workers.

Stand-alone users can easily request deployments exceeding the Compute service default maximum concurrent builds. As such, if your environment is used this way, you may wish to carefully consider your deployment architecture.

With a single `nova-compute` process talking to a single conductor, asked to perform ten concurrent deployments of images requiring conversion, the memory needed may exceed 10GB. This does however, entirely depend upon image block structure and layout, and what deploy interface is being used.

Database

Query load upon the database is one of the biggest potential bottlenecks which can cascade across a deployment and ultimately degrade service to an Ironic user.

Often, depending on load, query patterns, periodic tasks, and so on and so forth, additional indexes may be needed to help provide hints to the database so it can most efficiently attempt to reduce the number of rows which need to be examined in order to return a result set.

Adding indexes

This example below is specific to MariaDB/MySQL, but the syntax should be easy to modify for operators using PostgreSQL.

```
use ironic;
create index owner_idx on nodes (owner) LOCK = SHARED;
create index lessee_idx on nodes (lessee) LOCK = SHARED;
create index driver_idx on nodes (driver) LOCK = SHARED;
create index provision_state_idx on nodes (provision_state) LOCK = SHARED;
create index reservation_idx on nodes (reservation) LOCK = SHARED;
create index conductor_group_idx on nodes (conductor_group) LOCK = SHARED;
create index resource_class_idx on nodes (resource_class) LOCK = SHARED;
```

Note: The indexes noted have been added automatically by Xena versions of Ironic and later. They are provided here as an example and operators can add them manually prior with versions of Ironic. The database upgrade for the Xena release of Ironic which adds these indexes are only aware of being able to skip index creation if it already exists on MySQL/MariaDB.

Note: It may be possible to use LOCK = NONE. Basic testing indicates this takes a little bit longer, but shouldn't result in the database table becoming write locked during the index creation. If the database engine cannot support this, then the index creation will fail.

Database platforms also have a concept of what is called a compound index where the index is aligned with the exact query pattern being submitted to the database. The database is able to use this compound index to attempt to drastically reduce the result set generation time for the remainder of the query. As of the composition of this document, we do not ship compound indexes in Ironic as we feel the most general benefit is single column indexes, and depending on data present, an operator may wish to explore compound indexes with their database administrator, as compound indexes can also have negative performance impacts if improperly constructed.

```
use ironic;
create index my_custom_app_query_index on nodes (reservation, provision_state,
→ driver);
```

The risk, and *WHY* you should engage a Database Administrator, is depending on your configuration, the actual index may need to include one or more additional fields such as owner or lessee which may be added on to the index. At the same time, queries with less field matches, or in different orders will exhibit different performance as the compound index may not be able to be consulted.

Indexes will not fix everything

Indexes are not a magical cure-all for all API or database performance issues, but they are an incredibly important part depending on data access and query patterns.

The underlying object layer and data conversions including record pagination do add a substantial amount of overhead to what may otherwise return as a result set on a manual database query. In Ironics case, due to the object model and the need to extract multiple pieces of data at varying levels of the data model to handle cases such as upgrades, the entire result set is downloaded and transformed which is an overhead you do not experience with a command line database client.

What can I do?

Previously in this document, weve already suggested some architectural constraints and limitations, but there are some things that can be done to maximize performance. Again, this will vary greatly depending on your use.

- Use the `direct` deploy interface. This offloads any final image conversion to the host running the `ironic-python-agent`. Additionally, if Swift or other object storage such as RadosGW is used, downloads can be completely separated from the host running the `ironic-conductor`.
- Use small/compact raw images. Qcow2 files are generally compressed and require substantial amounts of memory to decompress and stream.
- Tune the internal memory limit for the conductor using the `[DEFAULT]memory_required_minimum` setting. This will help the conductor throttle back memory intensive operations. The default should prevent Out-of-Memory operations, but under extreme memory pressure this may still be sub-optimal. Before changing this setting, it is highly advised to consult with your resident Unix wizard or even the Ironic development team in upstream IRC. This feature was added in the Wallaby development cycle.
- If network bandwidth is the problem you are seeking to solve for, you may wish to explore a mix of the `direct` deploy interface and caching proxies. Such a configuration can be highly beneficial in wide area deployments. See *Using proxies for image download*.

Secure RBAC

Suggested Reading

It is likely an understatement to say that policy enforcement is a complex subject. It requires operational context to craft custom policy to meet general use needs. Part of this is why the Secure RBAC effort was started, to provide consistency and a good starting place for most users who need a higher level of granularity.

That being said, it would likely help anyone working to implement customization of these policies to consult some reference material in hopes of understanding the context.

- [Keystone Administrator Guide - Service API Protection](#)
- [Ironic Scoped Role Based Access Control Specification](#)

Historical Context - How we reached our access model

Ironic has reached the access model through an evolution the API and the data stored. Along with the data stored, the enforcement of policy based upon data stored in these fields.

- Ownership Information Storage
- Allow Node owners to Administer
- Allow Leasable Nodes

System Scoped

System scoped authentication is intended for administrative activities such as those crossing tenants/projects, as all tenants/projects should be visible to system scoped users in Ironic.

System scoped requests do not have an associated `project_id` value for the Keystone request authorization token utilized to speak with Ironic. These requests are translated through `keystonemiddleware` into values which tell Ironic what to do. Or to be more precise, tell the policy enforcement framework the information necessary to make decisions.

System scoped requests very much align with the access controls of Ironic before the Secure RBAC effort. The original custom role `baremetal_admin` privileges are identical to a system scoped `admins` privileges. Similarly `baremetal_reader` is identical to a system scoped `reader`. In these concepts, the `admin` is allowed to create/delete objects/items. The `reader` is allowed to read details about items and is intended for users who may need an account with read-only access for or front-line support purposes.

In addition to these concepts, a `member` role exists in the Secure RBAC use model. Ironic does support this role, and in general `member` role users in a system scope are able to perform basic updates/changes, with the exception of special fields like those to disable cleaning.

Project Scoped

Project scoped authentication is when a request token and associated records indicate an associated `project_id` value.

Legacy Behavior

The legacy behavior of API service is that all requests are treated as project scoped requests where access is governed using an admin project. This behavior is *deprecated*. The new behavior is a delineation of access through system scoped and project scoped requests.

In essence, what would have served as an admin project, is now system scoped usage.

Previously, Ironic API, by default, responded with access denied or permitted based upon the admin project and associated role. These responses would generate an HTTP 403 if the project was incorrect or if a user role.

Note: While Ironic has had the concept of an `owner` and a `lessee`, they are *NOT* used by default. They require custom policy configuration files to be used in the legacy operating mode.

Supported Endpoints

- /nodes
- /nodes/<uuid>/ports
- /nodes/<uuid>/portgroups
- /nodes/<uuid>/volume/connectors
- /nodes/<uuid>/volume/targets
- /nodes/<uuid>/allocation
- /ports
- /portgroups
- /volume/connectors
- /volume/targets
- /allocations

How Project Scoped Works

Ironic has two project use models where access is generally more delegative to an owner where access to a lessee is generally more utilitarian.

The purpose of an owner, is more to enable the System Operator to delegate much of the administrative activity of a Node to the owner. This may be because they physically own the hardware, or they are in charge of the node. Regardless of the use model that the fields and mechanics support, these fields are to support humans, and possibly services where applicable.

The purpose of a lessee is more for a *tenant* in their *project* to be able to have access to perform basic actions with the API. In some cases that may be to reprovision or rebuild a node. Ultimately that is the lessees prerogative, but by default there are actions and field updates that cannot be performed by default. This is also governed by access level within a project.

These policies are applied in the way data is viewed and how data can be updated. Generally, an inability to view a node is an access permission issue in term of the project ID being correct for owner/lessee.

The ironic project has attempted to generally codify what we believe is reasonable, however operators may wish to override these policy settings. For details general policy setting details, please see *Policies*.

Field value visibility restrictions

Ironics API, by default has a concept of filtering node values to prevent sensitive data from being leaked. System scoped users are subjected to basic restrictions, where as project scoped users are, by default, examined further and against additional policies. This threshold is controlled with the `baremetal:node:get:filter_threshold`.

By default, the following fields are masked on Nodes and are controlled by the associated policies. By default, owners are able to see insight into the infrastructure, where as lessee users *CANNOT* view these fields by default.

- `last_error` - `baremetal:node:get:last_error`

- reservation - baremetal:node:get:reservation
- driver_internal_info - baremetal:node:get:driver_internal_info
- driver_info - baremetal:node:get:driver_info

Field update restrictions

Some of the fields in this list are restricted to System scoped users, or even only System Administrators. Some of these default restrictions are likely obvious. Owners cant change the owner. Lessees cant change the owner.

- driver_info - baremetal:node:update:driver_info
- properties - baremetal:node:update:properties
- chassis_uuid - baremetal:node:update:chassis_uuid
- instance_uuid - baremetal:node:update:instance_uuid
- lessee - baremetal:node:update:lessee
- owner - baremetal:node:update:owner
- driver - baremetal:node:update:driver_interfaces
- *_interface - baremetal:node:update:driver_interfaces
- network_data - baremetal:node:update:network_data
- conductor_group - baremetal:node:update:conductor_group
- name - baremetal:node:update:name
- retired - baremetal:node:update:driver_info
- retired_reason - baremetal:node:update:retired

Warning: The chassis_uuid field is a write-once-only field. As such it is restricted to system scoped administrators.

More information is available on these fields in [Policies](#).

Allocations

The allocations endpoint of the API is somewhat different than other other endpoints as it allows for the allocation of physical machines to an admin. In this context, there is not already an owner or project_id to leverage to control access for the creation process, any project member does have the inherent privilege of requesting an allocation. That being said, their allocation request will require physical nodes to be owned or leased to the project_id through the node fields owner or lessee.

Ability to override the owner is restricted to system scoped users by default and any new allocation being requested with a specific owner, if made in project scope, will have the project_id recorded as the owner of the allocation.

Ultimately, an operational behavior difference exists between the owner and lessee rights in terms of allocations exists. With the standard access rights, lessee users are able to create allocations if they

own nodes which are not allocated or deployed, but they cannot reprovision nodes when using only a member role. This limitation is not the case for project-scoped users with the admin role.

Warning: The allocation endpoints use is restricted to project scoped interactions until `[oslo_policy]enforce_new_defaults` has been set to `True` using the `baremetal:allocation:create_pre_rbac` policy rule. This is in order to prevent endpoint misuse. Afterwards all project scoped allocations will automatically populate an owner. System scoped request are not subjected to this restriction, and operators may change the default restriction via the `baremetal:allocation:create_restricted` policy.

Practical differences

Most users, upon implementing the use of system scoped authentication should not notice a difference as long as their authentication token is properly scoped to `system` and with the appropriate role for their access level. For most users who used a `baremetal` project, or other custom project via a custom policy file, along with a custom role name such as `baremetal_admin`, this will require changing the user to be a system scoped user with admin privileges.

The most noticeable difference for API consumers is the HTTP 403 access code is now mainly a HTTP 404 access code. The access concept has changed from Does the user broadly has access to the API? to Does user have access to the node, and then do they have access to the specific resource?.

What is an owner or lessee?

An owner or lessee is the project which has been assigned baremetal resources. Generally these should be service projects as opposed to a project dedicated to a specific user. This will help prevent the need to involve a system scoped administrator from having to correct ownership records should a project need to be removed due to an individuals departure.

The underlying `project_id` is used to represent and associate the owner or lessee.

How do I assign an owner?

```
# baremetal node set --owner <project_id> <node>
```

Note: With the default access policy, an owner is able to change the assigned lessee of a node. However the lessee is unable to do the same.

How do I assign a lessee?

```
# baremetal node set --lessee <project_id> <node>
```

What is the difference between an owner and lessee?

This is largely covered in *How Project Scoped Works* although as noted it is largely in means of access. A lessee is far more restrictive and an owner may revoke access to lessee.

Access to the underlying baremetal node is not exclusive between the owner and lessee, and this use model expects that some level of communication takes place between the appropriate parties.

Deploy Steps

The deploy steps section has moved to *Node Deployment*.

4.2.24 Dashboard Integration

A plugin for the OpenStack Dashboard (horizon) service is under development. Documentation for that can be found within the ironic-ui project.

- [Dashboard \(horizon\) plugin](#)

CONFIGURATION GUIDE

5.1 Configuration Reference

Many aspects of the Bare Metal service are specific to the environment it is deployed in. The following pages describe configuration options that can be used to adjust the service to your particular situation.

5.1.1 Configuration Options

The following is an overview of all available configuration options in Ironic. For a sample configuration file, refer to *Sample Configuration File*.

DEFAULT

debug

Type boolean

Default False

Mutable This option can be changed without restarting.

If set to true, the logging level will be set to DEBUG instead of the default INFO level.

log_config_append

Type string

Default <None>

Mutable This option can be changed without restarting.

The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Table 1: Deprecated Variations

Group	Name
DEFAULT	log-config
DEFAULT	log_config

log_date_format

Type string

Default %Y-%m-%d %H:%M:%S

Defines the format string for `%(asctime)s` in log records. Default: the value above . This option is ignored if `log_config_append` is set.

log_file

Type string

Default <None>

(Optional) Name of log file to send logging output to. If no default is set, logging will go to `stderr` as defined by `use_stderr`. This option is ignored if `log_config_append` is set.

Table 2: Deprecated Variations

Group	Name
DEFAULT	logfile

log_dir

Type string

Default <None>

(Optional) The base directory used for relative `log_file` paths. This option is ignored if `log_config_append` is set.

Table 3: Deprecated Variations

Group	Name
DEFAULT	logdir

watch_log_file

Type boolean

Default False

Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if `log_file` option is specified and Linux platform is used. This option is ignored if `log_config_append` is set.

use_syslog

Type boolean

Default False

Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if `log_config_append` is set.

use_journal

Type boolean

Default False

Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if `log_config_append` is set.

syslog_log_facility

Type string

Default LOG_USER

Syslog facility to receive log lines. This option is ignored if `log_config_append` is set.

use_json

Type boolean

Default False

Use JSON formatting for logging. This option is ignored if `log_config_append` is set.

use_stderr

Type boolean

Default False

Log output to standard error. This option is ignored if `log_config_append` is set.

use_eventlog

Type boolean

Default False

Log output to Windows Event Log.

log_rotate_interval

Type integer

Default 1

The amount of time before the log files are rotated. This option is ignored unless `log_rotation_type` is set to `interval`.

log_rotate_interval_type

Type string

Default days

Valid Values Seconds, Minutes, Hours, Days, Weekday, Midnight

Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.

max_logfile_count

Type integer

Default 30

Maximum number of rotated log files.

max_logfile_size_mb

Type integer

Default 200

Log file maximum size in MB. This option is ignored if `log_rotation_type` is not set to `size`.

log_rotation_type

Type string

Default none

Valid Values interval, size, none

Log rotation type.

Possible values

interval Rotate logs at predefined time intervals.

size Rotate logs once they reach a predefined size.

none Do not rotate log files.

logging_context_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s
[%(request_id)s %(user_identity)s] %(instance)s%(message)s`

Format string to use for log messages with context. Used by `oslo_log.formatters.ContextFormatter`

logging_default_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s
[-] %(instance)s%(message)s`

Format string to use for log messages when context is undefined. Used by `oslo_log.formatters.ContextFormatter`

logging_debug_format_suffix

Type string

Default `%(funcName)s %(pathname)s:%(lineno)d`

Additional data to append to log message when logging level for the message is `DEBUG`. Used by `oslo_log.formatters.ContextFormatter`

logging_exception_prefix

Type string

Default `%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s
%(instance)s`

Prefix each line of exception output with this format. Used by `oslo_log.formatters.ContextFormatter`

logging_user_identity_format

Type string

Default %(user)s %(tenant)s %(domain)s %(user_domain)s
%(project_domain)s

Defines the format string for %(user_identity)s that is used in logging_context_format_string.
Used by oslo_log.formatters.ContextFormatter

default_log_levels

Type list

Default ['amqp=WARNING', 'amqpplib=WARNING', 'qpid.
messaging=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO',
'sqlalchemy=WARNING', 'stevedore=INFO', 'eventlet.wsgi.
server=INFO', 'iso8601=WARNING', 'requests=WARNING',
'glanceclient=WARNING', 'urllib3.connectionpool=WARNING',
'keystonemiddleware.auth_token=INFO', 'keystoneauth.
session=INFO', 'openstack=WARNING']

List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

publish_errors

Type boolean

Default False

Enables or disables publication of error events.

instance_format

Type string

Default "[instance: %(uuid)s] "

The format for an instance that is passed with the log message.

instance_uuid_format

Type string

Default "[instance: %(uuid)s] "

The format for an instance UUID that is passed with the log message.

rate_limit_interval

Type integer

Default 0

Interval, number of seconds, of log rate limiting.

rate_limit_burst

Type integer

Default 0

Maximum number of logged messages per rate_limit_interval.

rate_limit_except_level

Type string

Default CRITICAL

Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to `rate_limit_except_level` are not filtered. An empty string means that all levels are filtered.

fatal_deprecations

Type boolean

Default False

Enables or disables fatal status of deprecations.

rpc_conn_pool_size

Type integer

Default 30

Minimum Value 1

Size of RPC connection pool.

Table 4: Deprecated Variations

Group	Name
DEFAULT	rpc_conn_pool_size

conn_pool_min_size

Type integer

Default 2

The pool size limit for connections expiration policy

conn_pool_ttl

Type integer

Default 1200

The time-to-live in sec of idle connections in the pool

executor_thread_pool_size

Type integer

Default 64

Size of executor thread pool when executor is threading or eventlet.

Table 5: Deprecated Variations

Group	Name
DEFAULT	rpc_thread_pool_size

rpc_response_timeout

Type integer

Default 60

Seconds to wait for a response from a call.

transport_url

Type string

Default rabbit://

The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is:

```
driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query
```

Example: rabbit://rabbitmq:password@127.0.0.1:5672//

For full details on the fields in the URL see the documentation of `oslo_messaging.TransportURL` at <https://docs.openstack.org/oslo.messaging/latest/reference/transport.html>

control_exchange

Type string

Default openstack

The default exchange under which topics are scoped. May be overridden by an exchange name specified in the `transport_url` option.

rpc_ping_enabled

Type boolean

Default False

Add an endpoint to answer to ping calls. Endpoint is named `oslo_rpc_server_ping`

run_external_periodic_tasks

Type boolean

Default True

Some periodic tasks can be run in a separate process. Should we run them here?

backdoor_port

Type string

Default <None>

Enable eventlet backdoor. Acceptable values are 0, <port>, and <start>:<end>, where 0 results in listening on a random tcp port number; <port> results in listening on the specified port number (and not enabling backdoor if that port is in use); and <start>:<end> results in listening on the smallest unused port number within the specified range of port numbers. The chosen port is displayed in the services log file.

backdoor_socket

Type string

Default <None>

Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with `backdoor_port` in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path `{pid}` will be replaced with the PID of the current process.

log_options

Type boolean

Default True

Enables or disables logging values of all registered options when starting a service (at DEBUG level).

graceful_shutdown_timeout

Type integer

Default 60

Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.

auth_strategy

Type string

Default keystone

Valid Values noauth, keystone, http_basic

Authentication strategy used by ironic-api. noauth should not be used in a production environment because all authentication will be disabled.

Possible values

noauth no authentication

keystone use the Identity service for authentication

http_basic HTTP basic authentication

http_basic_auth_user_file

Type string

Default /etc/ironic/htpasswd

Path to Apache format user authentication file used when auth_strategy=http_basic

debug_tracebacks_in_api

Type boolean

Default False

Return server tracebacks in the API response for any error responses. **WARNING:** this is insecure and should not be used in a production environment.

pecan_debug

Type boolean

Default False

Enable pecan debug mode. **WARNING:** this is insecure and should not be used in a production environment.

default_resource_class

Type string

Default <None>

Mutable This option can be changed without restarting.

Resource class to use for new nodes when no resource class is provided in the creation request.

enabled_hardware_types

Type list

Default ['ipmi']

Specify the list of hardware types to load during service initialization. Missing hardware types, or hardware types which fail to initialize, will prevent the conductor service from starting. This option defaults to a recommended set of production-oriented hardware types. A complete list of hardware types present on your system may be found by enumerating the `ironic.hardware.types` endpoint.

enabled_bios_interfaces

Type list

Default ['no-bios']

Specify the list of bios interfaces to load during service initialization. Missing bios interfaces, or bios interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one bios interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented bios interfaces. A complete list of bios interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.bios` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled bios interfaces on every ironic-conductor service.

default_bios_interface

Type string

Default <None>

Default bios interface to be used for nodes that do not have `bios_interface` field set. A complete list of bios interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.bios` endpoint.

enabled_boot_interfaces

Type list

Default ['pxe']

Specify the list of boot interfaces to load during service initialization. Missing boot interfaces, or boot interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one boot interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented boot interfaces. A complete list of boot interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.boot` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled boot interfaces on every ironic-conductor service.

default_boot_interface

Type string

Default <None>

Default boot interface to be used for nodes that do not have `boot_interface` field set. A complete list of boot interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.boot` endpoint.

enabled_console_interfaces

Type list

Default ['no-console']

Specify the list of console interfaces to load during service initialization. Missing console interfaces, or console interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one console interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented console interfaces. A complete list of console interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.console` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled console interfaces on every `ironic-conductor` service.

default_console_interface

Type string

Default <None>

Default console interface to be used for nodes that do not have `console_interface` field set. A complete list of console interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.console` endpoint.

enabled_deploy_interfaces

Type list

Default ['direct']

Specify the list of deploy interfaces to load during service initialization. Missing deploy interfaces, or deploy interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one deploy interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented deploy interfaces. A complete list of deploy interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.deploy` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled deploy interfaces on every `ironic-conductor` service.

default_deploy_interface

Type string

Default <None>

Default deploy interface to be used for nodes that do not have `deploy_interface` field set. A complete list of deploy interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.deploy` endpoint.

enabled_inspect_interfaces

Type list

Default ['no-inspect']

Specify the list of inspect interfaces to load during service initialization. Missing inspect interfaces, or inspect interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one inspect interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented inspect interfaces. A complete list of inspect interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.inspect` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled inspect interfaces on every ironic-conductor service.

default_inspect_interface

Type string

Default <None>

Default inspect interface to be used for nodes that do not have `inspect_interface` field set. A complete list of inspect interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.inspect` entrypoint.

enabled_management_interfaces

Type list

Default ['ipmitool']

Specify the list of management interfaces to load during service initialization. Missing management interfaces, or management interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one management interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented management interfaces. A complete list of management interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.management` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled management interfaces on every ironic-conductor service.

default_management_interface

Type string

Default <None>

Default management interface to be used for nodes that do not have `management_interface` field set. A complete list of management interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.management` entrypoint.

enabled_network_interfaces

Type list

Default ['flat', 'noop']

Specify the list of network interfaces to load during service initialization. Missing network interfaces, or network interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one network interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented network interfaces. A

complete list of network interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.network` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled network interfaces on every ironic-conductor service.

default_network_interface

Type string

Default <None>

Default network interface to be used for nodes that do not have `network_interface` field set. A complete list of network interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.network` endpoint.

enabled_power_interfaces

Type list

Default ['ipmitool']

Specify the list of power interfaces to load during service initialization. Missing power interfaces, or power interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one power interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented power interfaces. A complete list of power interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.power` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled power interfaces on every ironic-conductor service.

default_power_interface

Type string

Default <None>

Default power interface to be used for nodes that do not have `power_interface` field set. A complete list of power interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.power` endpoint.

enabled_raid_interfaces

Type list

Default ['agent', 'no-raid']

Specify the list of raid interfaces to load during service initialization. Missing raid interfaces, or raid interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one raid interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented raid interfaces. A complete list of raid interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.raid` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled raid interfaces on every ironic-conductor service.

default_raid_interface

Type string

Default <None>

Default raid interface to be used for nodes that do not have `raid_interface` field set. A complete list of raid interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.raid` endpoint.

enabled_rescue_interfaces

Type list

Default ['no-rescue']

Specify the list of rescue interfaces to load during service initialization. Missing rescue interfaces, or rescue interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one rescue interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented rescue interfaces. A complete list of rescue interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.rescue` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled rescue interfaces on every `ironic-conductor` service.

default_rescue_interface

Type string

Default <None>

Default rescue interface to be used for nodes that do not have `rescue_interface` field set. A complete list of rescue interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.rescue` endpoint.

enabled_storage_interfaces

Type list

Default ['cinder', 'noop']

Specify the list of storage interfaces to load during service initialization. Missing storage interfaces, or storage interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one storage interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented storage interfaces. A complete list of storage interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.storage` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled storage interfaces on every `ironic-conductor` service.

default_storage_interface

Type string

Default noop

Default storage interface to be used for nodes that do not have `storage_interface` field set. A complete list of storage interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.storage` endpoint.

enabled_vendor_interfaces

Type list

Default ['ipmitool', 'no-vendor']

Specify the list of vendor interfaces to load during service initialization. Missing vendor interfaces, or vendor interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one vendor interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented vendor interfaces. A complete list of vendor interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.vendor` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled vendor interfaces on every ironic-conductor service.

default_vendor_interface

Type string

Default <None>

Default vendor interface to be used for nodes that do not have `vendor_interface` field set. A complete list of vendor interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.vendor` entrypoint.

log_in_db_max_size

Type integer

Default 4096

Max number of characters of any node `last_error/maintenance_reason` pushed to database.

hash_partition_exponent

Type integer

Default 5

Exponent to determine number of hash partitions to use when distributing load across conductors. Larger values will result in more even distribution of load and less load when rebalancing the ring, but more memory usage. Number of partitions per conductor is $(2^{\text{hash_partition_exponent}})$. This determines the granularity of rebalancing: given 10 hosts, and an exponent of the 2, there are 40 partitions in the ring. A few thousand partitions should make rebalancing smooth in most cases. The default is suitable for up to a few hundred conductors. Configuring for too many partitions has a negative impact on CPU usage.

hash_ring_reset_interval

Type integer

Default 15

Time (in seconds) after which the hash ring is considered outdated and is refreshed on the next access.

hash_ring_algorithm

Type string

Default md5

Valid Values sha3_256, md5, sha256, sha3_384, blake2b, shake_256, sha224, sha512, blake2s, sha3_512, shake_128, sha3_224, sha1, sha384

Advanced Option Intended for advanced users and not used by the majority of users, and might have a significant effect on stability and/or performance.

Hash function to use when building the hash ring. If running on a FIPS system, do not use md5. **WARNING:** all ironic services in a cluster **MUST** use the same algorithm at all times. Changing the algorithm requires an offline update.

force_raw_images

Type boolean

Default True

Mutable This option can be changed without restarting.

If True, convert backing images to raw disk image format.

raw_image_growth_factor

Type floating point

Default 2.0

Minimum Value 1.0

The scale factor used for estimating the size of a raw image converted from compact image formats such as QCOW2. Default is 2.0, must be greater than 1.0.

isolinux_bin

Type string

Default /usr/lib/syslinux/isolinux.bin

Path to isolinux binary file.

isolinux_config_template

Type string

Default \$pybasedir/common/isolinux_config.template

Template file for isolinux configuration file.

grub_config_path

Type string

Default /boot/grub/grub.cfg

GRUB2 configuration file location on the UEFI ISO images produced by ironic. The default value is usually incorrect and should not be relied on. If you use a GRUB2 image from a certain distribution, use a distribution-specific path here, e.g. EFI/ubuntu/grub.cfg

grub_config_template

Type string

Default \$pybasedir/common/grub_conf.template

Template file for grub configuration file.

ldlinux_c32

Type string

Default <None>

Path to `ldlinux.c32` file. This file is required for `syslinux 5.0` or later. If not specified, the file is looked for in `/usr/lib/syslinux/modules/bios/ldlinux.c32` and `/usr/share/syslinux/ldlinux.c32`.

esp_image

Type string

Default <None>

Path to EFI System Partition image file. This file is recommended for creating UEFI bootable ISO images efficiently. ESP image should contain a FAT12/16/32-formatted file system holding EFI boot loaders (e.g. GRUB2) for each hardware architecture ironic needs to boot. This option is only used when neither ESP nor ISO deploy image is configured to the node being deployed in which case ironic will attempt to fetch ESP image from the configured location or extract ESP image from UEFI-bootable deploy ISO image.

parallel_image_downloads

Type boolean

Default False

Mutable This option can be changed without restarting.

Run image downloads and raw format conversions in parallel.

my_ip

Type string

Default 127.0.0.1

This option has a sample default set, which means that its actual default value may vary from the one documented above.

IPv4 address of this host. If unset, will determine the IP programmatically. If unable to do so, will use 127.0.0.1. NOTE: This field does accept an IPv6 address as an override for templates and URLs, however it is recommended that `[DEFAULT]my_ipv6` is used along with DNS names for service URLs for dual-stack environments.

my_ipv6

Type string

Default 2001:db8::1

This option has a sample default set, which means that its actual default value may vary from the one documented above.

IP address of this host using IPv6. This value must be supplied via the configuration and cannot be adequately programmatically determined like the `[DEFAULT]my_ip` parameter for IPv4.

notification_level

Type string

Default <None>

Valid Values debug, info, warning, error, critical

Specifies the minimum level for which to send notifications. If not set, no notifications will be sent. The default is for this option to be unset.

Possible values

debug debug level

info info level

warning warning level

error error level

critical critical level

versioned_notifications_topics

Type list

Default ['ironic_versioned_notifications']

Specifies the topics for the versioned notifications issued by Ironic.

The default value is fine for most deployments and rarely needs to be changed. However, if you have a third-party service that consumes versioned notifications, it might be worth getting a topic for that service. Ironic will send a message containing a versioned notification payload to each topic queue in this list.

The list of versioned notifications is visible in <https://docs.openstack.org/ironic/latest/admin/notifications.html>

pybasedir

Type string

Default /usr/lib/python/site-packages/ironic/ironic

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Directory where the ironic python module is installed.

bindir

Type string

Default \$pybasedir/bin

Directory where ironic binaries are installed.

state_path

Type string

Default \$pybasedir

Top-level directory for maintaining ironics state.

default_portgroup_mode

Type string

Default active-backup

Mutable This option can be changed without restarting.

Default mode for portgroups. Allowed values can be found in the linux kernel documentation on bonding: <https://www.kernel.org/doc/Documentation/networking/bonding.txt>.

host

Type string

Default localhost

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Name of this node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address. However, the node name must be valid within an AMQP key, and if using ZeroMQ (will be removed in the Stein release), a valid hostname, FQDN, or IP address.

pin_release_version

Type string

Default <None>

Valid Values wallaby, victoria, 9.2, 18.0, 17.0, 16.2, 16.1, 16.0, 15.1, 15.0, 14.0, 13.0, 12.2, 12.1, 12.0, 11.1, 11.0, 10.1, 10.0

Mutable This option can be changed without restarting.

Used for rolling upgrades. Setting this option downgrades (or pins) the Bare Metal API, the internal ironiC RPC communication, and the database objects to their respective versions, so they are compatible with older services. When doing a rolling upgrade from version N to version N+1, set (to pin) this to N. To unpin (default), leave it unset and the latest versions will be used.

Possible values

wallaby wallaby release

victoria victoria release

9.2 9.2 release

18.0 18.0 release

17.0 17.0 release

16.2 16.2 release

16.1 16.1 release

16.0 16.0 release

15.1 15.1 release

15.0 15.0 release

14.0 14.0 release

13.0 13.0 release

12.2 12.2 release

12.1 12.1 release

12.0 12.0 release

11.1 11.1 release

11.0 11.0 release

10.1 10.1 release

10.0 10.0 release

rpc_transport

Type string

Default oslo

Valid Values oslo, json-rpc

Which RPC transport implementation to use between conductor and API services

Possible values

oslo use oslo.messaging transport

json-rpc use JSON RPC transport

minimum_memory_warning_only

Type boolean

Default False

Mutable This option can be changed without restarting.

Setting to govern if Ironic should only warn instead of attempting to hold back the request in order to prevent the exhaustion of system memory.

minimum_required_memory

Type integer

Default 1024

Mutable This option can be changed without restarting.

Minimum memory in MiB for the system to have available prior to starting a memory intensive process on the conductor.

minimum_memory_wait_time

Type integer

Default 15

Mutable This option can be changed without restarting.

Seconds to wait between retries for free memory before launching the process. This, combined with `memory_wait_retries` allows the conductor to determine how long we should attempt to directly retry.

minimum_memory_wait_retries

Type integer

Default 6

Mutable This option can be changed without restarting.

Number of retries to hold onto the worker before failing or returning the thread to the pool if the conductor can automatically retry.

rootwrap_config

Type string

Default /etc/ironic/rootwrap.conf

Path to the rootwrap configuration file to use for running commands as root.

tempdir

Type string

Default /tmp

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Temporary working directory, default is Python temp dir.

webserver_verify_ca

Type string

Default True

Mutable This option can be changed without restarting.

CA certificates to be used for certificate verification. This can be either a Boolean value or a path to a CA_BUNDLE file. If set to True, the certificates present in the standard path are used to verify the host certificates. If set to False, the conductor will ignore verifying the SSL certificate presented by the host. If it's a path, conductor uses the specified certificate for SSL verification. If the path does not exist, the behavior is same as when this value is set to True i.e the certificates present in the standard path are used for SSL verification. Defaults to True.

webserver_connection_timeout

Type integer

Default 60

Connection timeout when accessing remote web servers with images.

agent

manage_agent_boot

Type boolean

Default True

Whether Ironic will manage booting of the agent ramdisk. If set to False, you will need to configure your mechanism to allow booting the agent ramdisk.

memory_consumed_by_agent

Type integer

Default 0

Mutable This option can be changed without restarting.

The memory size in MiB consumed by agent when it is booted on a bare metal node. This is used for checking if the image can be downloaded and deployed on the bare metal node after booting agent ramdisk. This may be set according to the memory consumed by the agent ramdisk image.

stream_raw_images

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether the agent ramdisk should stream raw images directly onto the disk or not. By streaming raw images directly onto the disk the agent ramdisk will not spend time copying the image to a tmpfs partition (therefore consuming less memory) prior to writing it to the disk. Unless the disk where the image will be copied to is really slow, this option should be set to True. Defaults to True.

post_deploy_get_power_state_retries

Type integer

Default 6

Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off.

post_deploy_get_power_state_retry_interval

Type integer

Default 5

Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.

agent_api_version

Type string

Default v1

API version to use for communicating with the ramdisk agent.

deploy_logs_collect

Type string

Default on_failure

Valid Values always, on_failure, never

Mutable This option can be changed without restarting.

Whether Ironic should collect the deployment logs on deployment failure (on_failure), always or never.

Possible values

always always collect the logs

on_failure only collect logs if there is a failure

never never collect logs

deploy_logs_storage_backend

Type string

Default local

Valid Values local, swift

Mutable This option can be changed without restarting.

The name of the storage backend where the logs will be stored.

Possible values

local store the logs locally

swift store the logs in Object Storage service

deploy_logs_local_path

Type string

Default /var/log/ironic/deploy

Mutable This option can be changed without restarting.

The path to the directory where the logs should be stored, used when the `deploy_logs_storage_backend` is configured to local.

deploy_logs_swift_container

Type string

Default ironic_deploy_logs_container

Mutable This option can be changed without restarting.

The name of the Swift container to store the logs, used when the `deploy_logs_storage_backend` is configured to swift.

deploy_logs_swift_days_to_expire

Type integer

Default 30

Mutable This option can be changed without restarting.

Number of days before a log object is marked as expired in Swift. If None, the logs will be kept forever or until manually deleted. Used when the `deploy_logs_storage_backend` is configured to swift.

image_download_source

Type string

Default http

Valid Values swift, http, local

Mutable This option can be changed without restarting.

Specifies whether direct deploy interface should try to use the image source directly or if ironic should cache the image on the conductor and serve it from ironics own http server.

Possible values

swift IPA ramdisk retrieves instance image from the Object Storage service.

http IPA ramdisk retrieves instance image from HTTP service served at conductor nodes.

local Same as http, but HTTP images are also cached locally, converted and served from the conductor

command_timeout

Type integer

Default 60

Mutable This option can be changed without restarting.

Timeout (in seconds) for IPA commands.

max_command_attempts

Type integer

Default 3

This is the maximum number of attempts that will be done for IPA commands that fails due to network problems.

command_wait_attempts

Type integer

Default 100

Number of attempts to check for asynchronous commands completion before timing out.

command_wait_interval

Type integer

Default 6

Number of seconds to wait for between checks for asynchronous commands completion.

neutron_agent_poll_interval

Type integer

Default 2

Mutable This option can be changed without restarting.

The number of seconds Neutron agent will wait between polling for device changes. This value should be the same as CONF.AGENT.polling_interval in Neutron configuration.

neutron_agent_max_attempts

Type integer

Default 100

Max number of attempts to validate a Neutron agent status before raising network error for a dead agent.

neutron_agent_status_retry_interval

Type integer

Default 10

Wait time in seconds between attempts for validating Neutron agent status.

require_tls

Type boolean

Default False

Mutable This option can be changed without restarting.

If set to True, callback URLs without `https://` will be rejected by the conductor.

certificates_path

Type string

Default /var/lib/ironic/certificates

Path to store auto-generated TLS certificates used to validate connections to the ramdisk.

verify_ca

Type string

Default True

Path to the TLS CA to validate connection to the ramdisk. Set to True to use the system default CA storage. Set to False to disable validation. Ignored when automatic TLS setup is used.

api_ca_file

Type string

Default <None>

Path to the TLS CA that is used to start the bare metal API. In some boot methods this file can be passed to the ramdisk.

anaconda

default_ks_template

Type string

Default \$pybasedir/drivers/modules/ks.cfg.template

Mutable This option can be changed without restarting.

kickstart template to use when no kickstart template is specified in the `instance_info` or the glance OS image.

ansible**ansible_extra_args****Type** string**Default** <None>

Extra arguments to pass on every invocation of Ansible.

verbosity**Type** integer**Default** <None>**Minimum Value** 0**Maximum Value** 4

Set ansible verbosity level requested when invoking ansible-playbook command. 4 includes detailed SSH session logging. Default is 4 when global debug is enabled and 0 otherwise.

ansible_playbook_script**Type** string**Default** ansible-playbook

Path to ansible-playbook script. Default will search the \$PATH configured for user running ironic-conductor process. Provide the full path when ansible-playbook is not in \$PATH or installed in not default location.

playbooks_path**Type** string**Default** \$pybasedir/drivers/modules/ansible/playbooks

Path to directory with playbooks, roles and local inventory.

config_file_path**Type** string**Default** \$pybasedir/drivers/modules/ansible/playbooks/ansible.cfg

Path to ansible configuration file. If set to empty, system default will be used.

post_deploy_get_power_state_retries**Type** integer**Default** 6**Minimum Value** 0

Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off. Value of 0 means do not retry on failure.

post_deploy_get_power_state_retry_interval**Type** integer**Default** 5

Minimum Value 0

Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.

extra_memory

Type integer

Default 10

Extra amount of memory in MiB expected to be consumed by Ansible-related processes on the node. Affects decision whether image will fit into RAM.

image_store_insecure

Type boolean

Default False

Skip verifying SSL connections to the image store when downloading the image. Setting it to True is only recommended for testing environments that use self-signed certificates.

image_store_cafile

Type string

Default <None>

Specific CA bundle to use for validating SSL connections to the image store. If not specified, CA available in the ramdisk will be used. Is not used by default playbooks included with the driver. Suitable for environments that use self-signed certificates.

image_store_certfile

Type string

Default <None>

Client cert to use for SSL connections to image store. Is not used by default playbooks included with the driver.

image_store_keyfile

Type string

Default <None>

Client key to use for SSL connections to image store. Is not used by default playbooks included with the driver.

default_username

Type string

Default ansible

Name of the user to use for Ansible when connecting to the ramdisk over SSH. It may be overridden by per-node `ansible_username` option in nodes `driver_info` field.

default_key_file

Type string

Default <None>

Absolute path to the private SSH key file to use by Ansible by default when connecting to the ramdisk over SSH. Default is to use default SSH keys configured for the user running the ironic-conductor service. Private keys with password must be pre-loaded into ssh-agent. It may be overridden by per-node `ansible_key_file` option in nodes `driver_info` field.

default_deploy_playbook

Type string

Default `deploy.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for deployment. It may be overridden by per-node `ansible_deploy_playbook` option in nodes `driver_info` field.

default_shutdown_playbook

Type string

Default `shutdown.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for graceful in-band shutdown of the node. It may be overridden by per-node `ansible_shutdown_playbook` option in nodes `driver_info` field.

default_clean_playbook

Type string

Default `clean.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for node cleaning. It may be overridden by per-node `ansible_clean_playbook` option in nodes `driver_info` field.

default_clean_steps_config

Type string

Default `clean_steps.yaml`

Path (relative to `$playbooks_path` or absolute) to the default auxiliary cleaning steps file used during the node cleaning. It may be overridden by per-node `ansible_clean_steps_config` option in nodes `driver_info` field.

default_python_interpreter

Type string

Default `<None>`

Absolute path to the python interpreter on the managed machines. It may be overridden by per-node `ansible_python_interpreter` option in nodes `driver_info` field. By default, ansible uses `/usr/bin/python`

api

host_ip

Type host address

Default 0.0.0.0

The IP address or hostname on which ironic-api listens.

port

Type port number

Default 6385

Minimum Value 0

Maximum Value 65535

The TCP port on which ironic-api listens.

max_limit

Type integer

Default 1000

Mutable This option can be changed without restarting.

The maximum number of items returned in a single response from a collection resource.

public_endpoint

Type string

Default <None>

Mutable This option can be changed without restarting.

Public URL to use when building the links to the API resources (for example, <https://ironic.rocks:6384>). If None the links will be built using the requests host URL. If the API is operating behind a proxy, you will want to change this to represent the proxy's URL. Defaults to None. Ignored when proxy headers parsing is enabled via `[oslo_middleware]enable_proxy_headers_parsing` option.

api_workers

Type integer

Default <None>

Number of workers for OpenStack Ironic API service. The default is equal to the number of CPUs available, but not more than 4. One worker is used if the CPU number cannot be detected.

enable_ssl_api

Type boolean

Default False

Enable the integrated stand-alone API to service requests via HTTPS instead of HTTP. If there is a front-end service performing HTTPS offloading from the service, this option should be False; note, you will want to enable proxy headers parsing with `[oslo_middleware]enable_proxy_headers_parsing` option or configure `[api]public_endpoint` option to set URLs in responses to the SSL terminated one.

restrict_lookup

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to restrict the lookup API to only nodes in certain states.

ramdisk_heartbeat_timeout

Type integer

Default 300

Mutable This option can be changed without restarting.

Maximum interval (in seconds) for agent heartbeats.

network_data_schema

Type string

Default `$pybasedir/api/controllers/v1/network-data-schema.json`

Schema for network data used by this deployment.

audit**enabled**

Type boolean

Default False

Enable auditing of API requests (for ironic-api service).

audit_map_file

Type string

Default `/etc/ironic/api_audit_map.conf`

Path to audit map file for ironic-api service. Used only when API audit is enabled.

ignore_req_list

Type string

Default ''

Comma separated list of Ironic REST API HTTP methods to be ignored during audit logging. For example: auditing will not be done on any GET or POST requests if this is set to GET,POST. It is used only when API audit is enabled.

cinder

action_retries

Type integer

Default 3

Number of retries in the case of a failed action (currently only used when detaching volumes).

action_retry_interval

Type integer

Default 5

Retry interval in seconds in the case of a failed action (only specific actions are retried).

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 6: Deprecated Variations

Group	Name
cinder	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 7: Deprecated Variations

Group	Name
cinder	tenant-id
cinder	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 8: Deprecated Variations

Group	Name
cinder	tenant-name
cinder	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

retries

Type integer

Default 3

Client retries in the case of a failed request connection.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default volumev3

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 9: Deprecated Variations

Group	Name
cinder	user-name
cinder	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with `min_version` and `max_version`

conductor

workers_pool_size

Type integer

Default 100

Minimum Value 3

The size of the workers greenthread pool. Note that 2 threads will be reserved by the conductor itself for handling heart beats and periodic tasks. On top of that, `sync_power_state_workers` will take up to 7 green threads with the default value of 8.

heartbeat_interval

Type integer

Default 10

Seconds between conductor heart beats.

heartbeat_timeout

Type integer

Default 60

Maximum Value 315576000

Mutable This option can be changed without restarting.

Maximum time (in seconds) since the last check-in of a conductor. A conductor is considered inactive when this time has been exceeded.

sync_power_state_interval

Type integer

Default 60

Interval between syncing the node power state to the database, in seconds. Set to 0 to disable syncing.

check_provision_state_interval

Type integer

Default 60

Minimum Value 0

Interval between checks of provision timeouts, in seconds. Set to 0 to disable checks.

check_rescue_state_interval

Type integer

Default 60

Minimum Value 1

Interval (seconds) between checks of rescue timeouts.

check_allocations_interval

Type integer

Default 60

Minimum Value 0

Interval between checks of orphaned allocations, in seconds. Set to 0 to disable checks.

deploy_callback_timeout

Type integer

Default 1800

Minimum Value 0

Timeout (seconds) to wait for a callback from a deploy ramdisk. Set to 0 to disable timeout.

force_power_state_during_sync

Type boolean

Default True

Mutable This option can be changed without restarting.

During `sync_power_state`, should the hardware power state be set to the state recorded in the database (True) or should the database be updated based on the hardware state (False).

power_state_sync_max_retries

Type integer

Default 3

During `sync_power_state` failures, limit the number of times Ironic should try syncing the hardware node power state with the node power state in DB

sync_power_state_workers

Type integer

Default 8

Minimum Value 1

The maximum number of worker threads that can be started simultaneously to sync nodes power states from the periodic task.

periodic_max_workers

Type integer

Default 8

Maximum number of worker threads that can be started simultaneously by a periodic task. Should be less than RPC thread pool size.

node_locked_retry_attempts

Type integer

Default 3

Number of attempts to grab a node lock.

node_locked_retry_interval

Type integer

Default 1

Seconds to sleep between node lock attempts.

send_sensor_data

Type boolean

Default False

Enable sending sensor data message via the notification bus

send_sensor_data_interval

Type integer

Default 600

Minimum Value 1

Seconds between conductor sending sensor data message to ceilometer via the notification bus.

send_sensor_data_workers

Type integer

Default 4

Minimum Value 1

The maximum number of workers that can be started simultaneously for send data from sensors periodic task.

send_sensor_data_wait_timeout

Type integer

Default 300

The time in seconds to wait for send sensors data periodic task to be finished before allowing periodic call to happen again. Should be less than send_sensor_data_interval value.

send_sensor_data_types

Type list

Default ['ALL']

List of comma separated meter types which need to be sent to Ceilometer. The default value, ALL, is a special value meaning send all the sensor data.

send_sensor_data_for_undeployed_nodes

Type boolean

Default False

The default for sensor data collection is to only collect data for machines that are deployed, however operators may desire to know if there are failures in hardware that is not presently in use. When set to true, the conductor will collect sensor information from all nodes when sensor data collection is enabled via the send_sensor_data setting.

sync_local_state_interval

Type integer

Default 180

When conductors join or leave the cluster, existing conductors may need to update any persistent local state as nodes are moved around the cluster. This option controls how often, in seconds, each conductor will check for nodes that it should take over. Set it to 0 (or a negative value) to disable the check entirely.

configdrive_swift_container

Type string

Default ironic_configdrive_container

Name of the Swift container to store config drive data. Used when configdrive_use_object_store is True.

configdrive_swift_temp_url_duration

Type integer

Default <None>

Minimum Value 60

The timeout (in seconds) after which a configdrive temporary URL becomes invalid. Defaults to deploy_callback_timeout if it is set, otherwise to 1800 seconds. Used when configdrive_use_object_store is True.

inspect_wait_timeout**Type** integer**Default** 1800**Minimum Value** 0

Timeout (seconds) for waiting for node inspection. 0 - unlimited.

automated_clean**Type** boolean**Default** True**Mutable** This option can be changed without restarting.

Enables or disables automated cleaning. Automated cleaning is a configurable set of steps, such as erasing disk drives, that are performed on the node to ensure it is in a baseline state and ready to be deployed to. This is done after instance deletion as well as during the transition from a manageable to available state. When enabled, the particular steps performed to clean a node depend on which driver that node is managed by; see the individual drivers documentation for details. NOTE: The introduction of the cleaning operation causes instance deletion to take significantly longer. In an environment where all tenants are trusted (eg, because there is only one tenant), this option could be safely disabled.

allow_provisioning_in_maintenance**Type** boolean**Default** True**Mutable** This option can be changed without restarting.

Whether to allow nodes to enter or undergo deploy or cleaning when in maintenance mode. If this option is set to False, and a node enters maintenance during deploy or cleaning, the process will be aborted after the next heartbeat. Automated cleaning or making a node available will also fail. If True (the default), the process will begin and will pause after the node starts heartbeating. Moving it from maintenance will make the process continue.

clean_callback_timeout**Type** integer**Default** 1800**Minimum Value** 0

Timeout (seconds) to wait for a callback from the ramdisk doing the cleaning. If the timeout is reached the node will be put in the clean failed provision state. Set to 0 to disable timeout.

rescue_callback_timeout**Type** integer**Default** 1800**Minimum Value** 0

Timeout (seconds) to wait for a callback from the rescue ramdisk. If the timeout is reached the node will be put in the rescue failed provision state. Set to 0 to disable timeout.

soft_power_off_timeout

Type integer

Default 600

Minimum Value 1

Mutable This option can be changed without restarting.

Timeout (in seconds) of soft reboot and soft power off operation. This value always has to be positive.

power_state_change_timeout

Type integer

Default 60

Minimum Value 2

Mutable This option can be changed without restarting.

Number of seconds to wait for power operations to complete, i.e., so that a baremetal node is in the desired power state. If timed out, the power operation is considered a failure.

power_failure_recovery_interval

Type integer

Default 300

Minimum Value 0

Interval (in seconds) between checking the power state for nodes previously put into maintenance mode due to power synchronization failure. A node is automatically moved out of maintenance mode once its power state is retrieved successfully. Set to 0 to disable this check.

conductor_group

Type string

Default ''

Name of the conductor group to join. Can be up to 255 characters and is case insensitive. This conductor will only manage nodes with a matching conductor_group field set on the node.

allow_deleting_available_nodes

Type boolean

Default True

Mutable This option can be changed without restarting.

Allow deleting nodes which are in state available. Defaults to True.

enable_mdns

Type boolean

Default False

Whether to enable publishing the baremetal API endpoint via multicast DNS.

deploy_kernel

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default deploy image.

deploy_ramdisk

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default deploy image.

rescue_kernel

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default rescue image.

rescue_ramdisk

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default rescue image.

rescue_password_hash_algorithm

Type string

Default sha256

Valid Values sha256, sha512

Mutable This option can be changed without restarting.

Password hash algorithm to be used for the rescue password.

require_rescue_password_hashed

Type boolean

Default False

Mutable This option can be changed without restarting.

Option to cause the conductor to not fallback to an un-hashed version of the rescue password, permitting rescue with older ironic-python-agent ramdisks.

bootloader

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, <http://> or <file://> URL of the EFI system partition image containing EFI boot loader. This image will be used by ironic when building UEFI-bootable ISO out of kernel and ramdisk. Required for UEFI boot from partition images.

clean_step_priority_override

Type unknown type

Default {}

Priority to run automated clean steps for both in-band and out of band clean steps, provided in interface.step_name:priority format, e.g. deploy.erase_devices_metadata:123. The option can be specified multiple times to define priorities for multiple steps. If set to 0, this specific step will not run during cleaning. If unset for an inband clean step, will use the priority set in the ramdisk.

console

terminal

Type string

Default shellinaboxd

Path to serial console terminal program. Used only by Shell In A Box console.

terminal_cert_dir

Type string

Default <None>

Directory containing the terminal SSL cert (PEM) for serial console access. Used only by Shell In A Box console.

terminal_pid_dir

Type string

Default <None>

Directory for holding terminal pid files. If not specified, the temporary directory will be used.

terminal_timeout

Type integer

Default 600

Minimum Value 0

Timeout (in seconds) for the terminal session to be closed on inactivity. Set to 0 to disable timeout. Used only by Socat console.

subprocess_checking_interval

Type integer

Default 1

Time interval (in seconds) for checking the status of console subprocess.

subprocess_timeout

Type integer

Default 10

Time (in seconds) to wait for the console subprocess to start.

kill_timeout

Type integer

Default 1

Time (in seconds) to wait for the console subprocess to exit before sending SIGKILL signal.

socat_address

Type ip address

Default \$my_ip

IP address of Socat service running on the host of ironic conductor. Used only by Socat console.

port_range

Type string

Default 10000:20000

This option has a sample default set, which means that its actual default value may vary from the one documented above.

A range of ports available to be used for the console proxy service running on the host of ironic conductor, in the form of <start>:<stop>. This option is used by both Shellinabox and Socat console

cors

allowed_origin

Type list

Default <None>

Indicate whether this resource may be shared with the domain received in the requests origin header. Format: <protocol>://<host>[:<port>], no trailing slash. Example: <https://horizon.example.com>

allow_credentials

Type boolean

Default True

Indicate that the actual request can include user credentials

expose_headers

Type list

Default []

Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.

max_age

Type integer

Default 3600

Maximum cache age of CORS preflight requests.

allow_methods

Type list

Default ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE', 'PATCH']

Indicate which methods can be used during the actual request.

allow_headers

Type list

Default []

Indicate which header field names may be used during the actual request.

database

sqlite_synchronous

Type boolean

Default True

If True, SQLite uses synchronous mode.

Table 10: Deprecated Variations

Group	Name
DEFAULT	sqlite_synchronous

backend

Type string

Default sqlalchemy

The back end to use for the database.

Table 11: Deprecated Variations

Group	Name
DEFAULT	db_backend

connection

Type string

Default <None>

The SQLAlchemy connection string to use to connect to the database.

Table 12: Deprecated Variations

Group	Name
DEFAULT	sql_connection
DATABASE	sql_connection
sql	connection

slave_connection**Type** string**Default** <None>

The SQLAlchemy connection string to use to connect to the slave database.

mysql_sql_mode**Type** string**Default** TRADITIONAL

The SQL mode to be used for MySQL sessions. This option, including the default, overrides any server-set SQL mode. To use whatever SQL mode is set by the server configuration, set this to no value. Example: `mysql_sql_mode=`

mysql_enable_ndb**Type** boolean**Default** False

If True, transparently enables support for handling MySQL Cluster (NDB).

connection_recycle_time**Type** integer**Default** 3600

Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.

max_pool_size**Type** integer**Default** 5

Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.

max_retries**Type** integer**Default** 10

Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Table 13: Deprecated Variations

Group	Name
DEFAULT	sql_max_retries
DATABASE	sql_max_retries

retry_interval**Type** integer**Default** 10

Interval between retries of opening a SQL connection.

Table 14: Deprecated Variations

Group	Name
DEFAULT	sql_retry_interval
DATABASE	reconnect_interval

max_overflow

Type integer

Default 50

If set, use this value for max_overflow with SQLAlchemy.

Table 15: Deprecated Variations

Group	Name
DEFAULT	sql_max_overflow
DATABASE	sqlalchemy_max_overflow

connection_debug

Type integer

Default 0

Minimum Value 0

Maximum Value 100

Verbosity of SQL debugging information: 0=None, 100=Everything.

Table 16: Deprecated Variations

Group	Name
DEFAULT	sql_connection_debug

connection_trace

Type boolean

Default False

Add Python stack traces to SQL as comment strings.

Table 17: Deprecated Variations

Group	Name
DEFAULT	sql_connection_trace

pool_timeout

Type integer

Default <None>

If set, use this value for `pool_timeout` with SQLAlchemy.

Table 18: Deprecated Variations

Group	Name
DATABASE	sqlalchemy_pool_timeout

use_db_reconnect

Type boolean

Default False

Enable the experimental use of database reconnect on connection lost.

db_retry_interval

Type integer

Default 1

Seconds between retries of a database transaction.

db_inc_retry_interval

Type boolean

Default True

If True, increases the interval between retries of a database operation up to `db_max_retry_interval`.

db_max_retry_interval

Type integer

Default 10

If `db_inc_retry_interval` is set, the maximum seconds between retries of a database operation.

db_max_retries

Type integer

Default 20

Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.

connection_parameters

Type string

Default ''

Optional URL parameters to append onto the connection URL at connect time; specify as `param1=value1¶m2=value2&`

mysql_engine

Type string

Default InnoDB

MySQL engine to use.

deploy

http_url

Type string

Default <None>

ironic-conductor nodes HTTP server URL. Example: <http://192.1.2.3:8080>

http_root

Type string

Default /httpboot

ironic-conductor nodes HTTP root path.

external_http_url

Type string

Default <None>

URL of the ironic-conductor nodes HTTP server for boot methods such as virtual media, where images could be served outside of the provisioning network. Does not apply when Swift is used. Defaults to http_url.

external_callback_url

Type string

Default <None>

Agent callback URL of the bare metal API for boot methods such as virtual media, where images could be served outside of the provisioning network. Defaults to the configuration from [service_catalog].

enable_ata_secure_erase

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to support the use of ATA Secure Erase during the cleaning process. Defaults to True.

enable_nvme_secure_erase

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to support the use of NVMe Secure Erase during the cleaning process. Currently nvmecli format command is supported with user-data and crypto modes, depending on device capabilities. Defaults to True.

erase_devices_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band erase devices via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 10 for the GenericHardwareManager). If set to 0, will not run during cleaning.

erase_devices_metadata_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that erases metadata from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 99 for the GenericHardwareManager). If set to 0, will not run during cleaning.

delete_configuration_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that erases RAID configuration from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 for the GenericHardwareManager). If set to 0, will not run during cleaning.

create_configuration_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that creates RAID configuration from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 for the GenericHardwareManager). If set to 0, will not run during cleaning.

shred_random_overwrite_iterations

Type integer

Default 1

Minimum Value 0

Mutable This option can be changed without restarting.

During shred, overwrite all block devices N times with random data. This is only used if a device could not be ATA Secure Erased. Defaults to 1.

shred_final_overwrite_with_zeros

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to write zeros to a nodes block devices after writing random data. This will write zeros to the device even when `deploy.shred_random_overwrite_iterations` is 0. This option is only used if a device could not be ATA Secure Erased. Defaults to True.

continue_if_disk_secure_erase_fails

Type boolean

Default False

Mutable This option can be changed without restarting.

Defines what to do if a secure erase operation (NVMe or ATA) fails during cleaning in the Ironic Python Agent. If False, the cleaning operation will fail and the node will be put in `clean failed` state. If True, `shred` will be invoked and cleaning will continue.

disk_erasure_concurrency

Type integer

Default 1

Minimum Value 1

Mutable This option can be changed without restarting.

Defines the target pool size used by Ironic Python Agent ramdisk to erase disk devices. The number of threads created to erase disks will not exceed this value or the number of disks to be erased.

power_off_after_deploy_failure

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to power off a node after deploy failure. Defaults to True.

default_boot_option

Type string

Default local

Valid Values netboot, local

Mutable This option can be changed without restarting.

Default boot option to use when no boot option is requested in nodes `driver_info`. Defaults to local. Prior to the Ussuri release, the default was netboot.

Possible values

netboot boot from a network

local local boot

default_boot_mode

Type string

Default bios

Valid Values uefi, bios

Mutable This option can be changed without restarting.

Default boot mode to use when no boot mode is requested in nodes `driver_info`, `capabilities` or in the `instance_info` configuration. Currently the default boot mode is bios, but it will be changed to uefi in the future. It is recommended to set an explicit value for this option. This option only has effect when management interface supports boot mode management

Possible values

uefi UEFI boot mode

bios Legacy BIOS boot mode

`configdrive_use_object_store`

Type boolean

Default False

Mutable This option can be changed without restarting.

Whether to upload the config drive to object store. Set this option to True to store config drive in a swift endpoint.

Table 19: Deprecated Variations

Group	Name
conductor	<code>configdrive_use_swift</code>

`http_image_subdir`

Type string

Default `agent_images`

The name of subdirectory under ironic-conductor nodes HTTP root path which is used to place instance images for the direct deploy interface, when local HTTP service is incorporated to provide instance image instead of swift tempurls.

`fast_track`

Type boolean

Default False

Mutable This option can be changed without restarting.

Whether to allow deployment agents to perform lookup, heartbeat operations during initial states of a machine lifecycle and by-pass the normal setup procedures for a ramdisk. This feature also enables power operations which are part of deployment processes to be bypassed if the ramdisk has performed a heartbeat operation using the `fast_track_timeout` setting.

`fast_track_timeout`

Type integer

Default 300

Minimum Value 0

Maximum Value 300

Mutable This option can be changed without restarting.

Seconds for which the last heartbeat event is to be considered valid for the purpose of a fast track sequence. This setting should generally be less than the number of seconds for Power-On Self Test and typical ramdisk start-up. This value should not exceed the `[api]ramdisk_heartbeat_timeout` setting.

erase_skip_read_only

Type boolean

Default False

Mutable This option can be changed without restarting.

If the `ironic-python-agent` should skip read-only devices when running the `erase_devices clean` step where block devices are zeroed out. This requires `ironic-python-agent 6.0.0` or greater. By default a read-only device will cause non-metadata based cleaning operations to fail due to the possible operational security risk of data being retained between deployments of the bare metal node.

ramdisk_image_download_source

Type string

Default local

Valid Values http, local

Mutable This option can be changed without restarting.

Specifies whether a boot iso image should be served from its own original location using the image source url directly, or if `ironic` should cache the image on the conductor and serve it from `ironics` own http server.

Possible values

http In case the ramdisk is already a bootable iso, using this option it will be directly provided by an external HTTP service using its full url.

local This is the default behavior. The image is downloaded, prepared and cached locally, to be served from the conductor.

dhcp

dhcp_provider

Type string

Default neutron

DHCP provider to use. `neutron` uses Neutron, and `none` uses a no-op provider.

disk_partitioner

check_device_interval

Type integer

Default 1

After Ironic has completed creating the partition table, it continues to check for activity on the attached iSCSI device status at this interval prior to copying the image to the node, in seconds

check_device_max_retries

Type integer

Default 20

The maximum number of times to check that the device is not accessed by another process. If the device is still busy after that, the disk partitioning will be treated as having failed.

disk_utils

efi_system_partition_size

Type integer

Default 200

Size of EFI system partition in MiB when configuring UEFI systems for local boot.

bios_boot_partition_size

Type integer

Default 1

Size of BIOS Boot partition in MiB when configuring GPT partitioned systems for local boot in BIOS.

dd_block_size

Type string

Default 1M

Block size to use when writing to the nodes disk.

partition_detection_attempts

Type integer

Default 3

Minimum Value 1

Maximum attempts to detect a newly created partition.

partprobe_attempts

Type integer

Default 10

Maximum number of attempts to try to read the partition.

image_convert_memory_limit

Type integer

Default 2048

Memory limit for qemu-img convert in MiB. Implemented via the address space resource limit.

image_convert_attempts

Type integer

Default 3

Number of attempts to convert an image.

drac

query_raid_config_job_status_interval

Type integer

Default 120

Minimum Value 1

Interval (in seconds) between periodic RAID job status checks to determine whether the asynchronous RAID configuration was successfully finished or not.

boot_device_job_status_timeout

Type integer

Default 30

Minimum Value 1

Maximum amount of time (in seconds) to wait for the boot device configuration job to transition to the correct state to allow a reboot or power on to complete.

config_job_max_retries

Type integer

Default 240

Minimum Value 1

Maximum number of retries for the configuration job to complete successfully.

query_import_config_job_status_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for completed import configuration task

bios_factory_reset_timeout

Type integer

Default 600

Minimum Value 1

Maximum time (in seconds) to wait for factory reset of BIOS settings to complete.

glance**allowed_direct_url_schemes**

Type list

Default []

A list of URL schemes that can be downloaded directly via the direct_url. Currently supported schemes: [file].

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 20: Deprecated Variations

Group	Name
glance	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

num_retries

Type integer

Default 0

Number of retries when downloading an image from glance.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 21: Deprecated Variations

Group	Name
glance	tenant-id
glance	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 22: Deprecated Variations

Group	Name
glance	tenant-name
glance	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default image

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

swift_account

Type string

Default <None>

The account that Glance uses to communicate with Swift. The format is AUTH_uuid. uuid is the UUID for the account configured in the glance-api.conf. For example: AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30. If not set, the default value is calculated based on the ID of the project used to access Swift (as set in the [swift] section). Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_api_version

Type string

Default v1

The Swift API version to create a temporary URL for. Defaults to v1. Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_container

Type string

Default glance

The Swift container Glance is configured to store its images in. Defaults to glance, which is the default in glance-api.conf. Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_endpoint_url

Type string

Default <None>

The endpoint (scheme, hostname, optional port) for the Swift URL of the form endpoint_url/api_version/account/container/object_id. Do not include trailing /. For example, use <https://swift.example.com>. If using RADOS Gateway, endpoint may also contain /swift path; if it does not, it will be appended. Used for temporary URLs, will be fetched from the service catalog, if not provided.

swift_store_multiple_containers_seed

Type integer

Default 0

This should match a config by the same name in the Glance configuration file. When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.

swift_temp_url_cache_enabled

Type boolean

Default False

Whether to cache generated Swift temporary URLs. Setting it to true is only useful when an image caching proxy is used. Defaults to False.

swift_temp_url_duration

Type integer

Default 1200

The length of time in seconds that the temporary URL will be valid for. Defaults to 20 minutes. If some deploys get a 401 response code when trying to download from the temporary URL, try raising this duration. This value must be greater than or equal to the value for `swift_temp_url_expected_download_start_delay`

swift_temp_url_expected_download_start_delay

Type integer

Default 0

Minimum Value 0

This is the delay (in seconds) from the time of the deploy request (when the Swift temporary URL is generated) to when the IPA ramdisk starts up and URL is used for the image download. This value is used to check if the Swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled this will determine if a cached entry will still be valid when the download starts. `swift_temp_url_duration` value must be greater than or equal to this options value. Defaults to 0.

swift_temp_url_key

Type string

Default <None>

The secret token given to Swift to allow temporary URL downloads. Required for temporary URLs. For the Swift backend, the key on the service project (as set in the [swift] section) is used by default.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout**Type** integer**Default** <None>

Timeout value for http requests

trust_id**Type** unknown type**Default** <None>

Trust ID

user_domain_id**Type** unknown type**Default** <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type**Default** <None>

User id

username**Type** unknown type**Default** <None>

Username

Table 23: Deprecated Variations

Group	Name
glance	user-name
glance	user_name

valid_interfaces**Type** list**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version**Type** string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with `min_version` and `max_version`

healthcheck

path

Type string

Default /healthcheck

The path to respond to healthcheck requests on.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

detailed

Type boolean

Default False

Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.

backends

Type list

Default []

Additional backends that can perform health checks and report that information back as part of a request.

disable_by_file_path

Type string

Default <None>

Check the presence of a file to determine if an application is running on a port. Used by `DisableByFileHealthcheck` plugin.

disable_by_file_paths

Type list

Default []

Check the presence of a file based on a port to determine if an application is running on a port. Expects a `port:path` list of strings. Used by `DisableByFilesPortsHealthcheck` plugin.

enabled

Type boolean

Default False

Enable the health check endpoint at /healthcheck. Note that this is unauthenticated. More information is available at https://docs.openstack.org/oslo.middleware/latest/reference/healthcheck_plugins.html.

ilo

client_timeout

Type integer

Default 60

Timeout (in seconds) for iLO operations

client_port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Port to be used for iLO operations

swift_ilo_container

Type string

Default ironic_ilo_container

The Swift iLO container to store data.

swift_object_expiry_timeout

Type integer

Default 900

Amount of time in seconds for Swift objects to auto-expire.

use_web_server_for_images

Type boolean

Default False

Set this to True to use http web server to host floppy images and generated boot ISO. This requires http_root and http_url to be configured in the [deploy] section of the config file. If this is set to False, then Ironic will use Swift to host the floppy images and generated boot_iso.

clean_priority_reset_ilo

Type integer

Default 0

Priority for reset_ilo clean step.

clean_priority_reset_bios_to_default

Type integer

Default 10

Priority for reset_bios_to_default clean step.

clean_priority_reset_secure_boot_keys_to_default

Type integer

Default 20

Priority for reset_secure_boot_keys clean step. This step will reset the secure boot keys to manufacturing defaults.

clean_priority_clear_secure_boot_keys

Type integer

Default 0

Priority for clear_secure_boot_keys clean step. This step is not enabled by default. It can be enabled to clear all secure boot keys enrolled with iLO.

clean_priority_reset_ilo_credential

Type integer

Default 30

Priority for reset_ilo_credential clean step. This step requires ilo_change_password parameter to be updated in nodess driver_info with the new password.

power_wait

Type integer

Default 2

Amount of time in seconds to wait in between power operations

oob_erase_devices_job_status_interval

Type integer

Default 300

Minimum Value 10

Interval (in seconds) between periodic erase-devices status checks to determine whether the asynchronous out-of-band erase-devices was successfully finished or not. On an average, a 300GB HDD with default pattern overwrite would take approximately 9 hours and 300GB SSD with default pattern block would take approx. 30 seconds to complete sanitize disk erase.

ca_file

Type string

Default <None>

CA certificate file to validate iLO.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Its being replaced by new configuration parameter verify_ca.

verify_ca**Type** string**Default** True

CA certificate to validate iLO. This can be either a Boolean value, a path to a CA_BUNDLE file or directory with certificates of trusted CAs. If set to True the driver will verify the host certificates; if False the driver will ignore verifying the SSL certificate. If its a path the driver will use the specified certificate or one of the certificates in the directory. Defaults to True.

default_boot_mode**Type** string**Default** auto**Valid Values** auto, bios, uefi

Default boot mode to be used in provisioning when boot_mode capability is not provided in the properties/capabilities of the node. The default is auto for backward compatibility. When auto is specified, default boot mode will be selected based on boot mode settings on the system.

Possible values

auto based on boot mode settings on the system

bios BIOS boot mode

uefi UEFI boot mode

file_permission**Type** integer**Default** 420

File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644, or as the octal number 0o644 in Python. This setting must be set to the octal number representation, meaning starting with 0o.

kernel_append_params**Type** string**Default** nofb nomodeset vga=normal**Mutable** This option can be changed without restarting.

Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading /proc/cmdline. Mind severe cmdline size limit! Can be overridden by *instance_info/kernel_append_params* property.

inspector

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 24: Deprecated Variations

Group	Name
inspector	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

callback_endpoint_override

Type string

Default <None>

endpoint to use as a callback for posting back introspection data when boot is managed by ironic. Standard keystoneauth options are used by default.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

extra_kernel_params

Type string

Default ''

extra kernel parameters to pass to the inspection ramdisk when boot is managed by ironic (not ironic-inspector). Pairs key=value separated by spaces.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

power_off

Type boolean

Default True

whether to power off a node after inspection finishes

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 25: Deprecated Variations

Group	Name
inspector	tenant-id
inspector	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 26: Deprecated Variations

Group	Name
inspector	tenant-name
inspector	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

require_managed_boot

Type boolean

Default False

require that the in-band inspection boot is fully managed by ironic. Set this to True if your installation of ironic-inspector does not have a separate PXE boot environment.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default baremetal-introspection

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

status_check_period

Type integer

Default 60

period (in seconds) to check status of nodes on inspection

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type**Default** <None>

User id

username**Type** unknown type**Default** <None>

Username

Table 27: Deprecated Variations

Group	Name
inspector	user-name
inspector	user_name

valid_interfaces**Type** list**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version**Type** string**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

ipmi**command_retry_timeout****Type** integer**Default** 60**Mutable** This option can be changed without restarting.

Maximum time in seconds to retry retryable IPMI operations. (An operation is retryable, for example, if the requested operation fails because the BMC is busy.) Setting this too high can cause the sync power state periodic task to hang when there are slow or unresponsive BMCs.

min_command_interval

Type integer

Default 5

Mutable This option can be changed without restarting.

Minimum time, in seconds, between IPMI operations sent to a server. There is a risk with some hardware that setting this too low may cause the BMC to crash. Recommended setting is 5 seconds.

use_ipmitool_retries

Type boolean

Default False

When set to True and the parameters are supported by ipmitool, the number of retries and the retry interval are passed to ipmitool as parameters, and ipmitool will do the retries. When set to False, ironic will retry the ipmitool commands. Recommended setting is False

kill_on_timeout

Type boolean

Default True

Mutable This option can be changed without restarting.

Kill *ipmitool* process invoked by ironic to read node power state if *ipmitool* process does not exit after *command_retry_timeout* timeout expires. Recommended setting is True

disable_boot_timeout

Type boolean

Default True

Mutable This option can be changed without restarting.

Default timeout behavior whether ironic sends a raw IPMI command to disable the 60 second timeout for booting. Setting this option to False will NOT send that command, the default value is True. It may be overridden by per-node *ipmi_disable_boot_timeout* option in nodes *driver_info* field.

additional_retryable_ipmi_errors

Type multi-valued

Default ''

Mutable This option can be changed without restarting.

Additional errors ipmitool may encounter, specific to the environment it is run in.

debug

Type boolean

Default False

Mutable This option can be changed without restarting.

Enables all ipmi commands to be executed with an additional debugging output. This is a separate option as ipmitool can log a substantial amount of misleading text when in this mode.

cipher_suite_versions

Type list

Default []

List of possible cipher suites versions that can be supported by the hardware in case the field *cipher_suite* is not set for the node.

irmc

remote_image_share_root

Type string

Default /remote_image_share_root

Ironic conductor nodes NFS or CIFS root path

remote_image_server

Type string

Default <None>

IP of remote image server

remote_image_share_type

Type string

Default CIFS

Valid Values CIFS, NFS

Share type of virtual media

Possible values

CIFS CIFS (Common Internet File System) protocol

NFS NFS (Network File System) protocol

remote_image_share_name

Type string

Default share

share name of remote_image_server

remote_image_user_name

Type string

Default <None>

User name of remote_image_server

remote_image_user_password

Type string

Default <None>

Password of remote_image_user_name

remote_image_user_domain

Type string

Default ''

Domain name of remote_image_user_name

port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Valid Values 443, 80

Port to be used for iRMC operations

Possible values

443 port 443

80 port 80

auth_method

Type string

Default basic

Valid Values basic, digest

Authentication method to be used for iRMC operations

Possible values

basic Basic authentication

digest Digest authentication

client_timeout

Type integer

Default 60

Timeout (in seconds) for iRMC operations

sensor_method**Type** string**Default** ipmitool**Valid Values** ipmitool, scci

Sensor data retrieval method.

Possible values**ipmitool** IPMItool**scci** Fujitsu SCCI (ServerView Common Command Interface)**snmp_version****Type** string**Default** v2c**Valid Values** v1, v2c, v3

SNMP protocol version

Possible values**v1** SNMPv1**v2c** SNMPv2c**v3** SNMPv3**snmp_port****Type** port number**Default** 161**Minimum Value** 0**Maximum Value** 65535

SNMP port

snmp_community**Type** string**Default** public

SNMP community. Required for versions v1 and v2c

snmp_security**Type** string**Default** <None>

SNMP security name. Required for version v3

snmp_polling_interval

Type integer

Default 10

SNMP polling interval in seconds

clean_priority_restore_irmc_bios_config

Type integer

Default 0

Priority for restore_irmc_bios_config clean step.

gpu_ids

Type list

Default []

List of vendor IDs and device IDs for GPU device to inspect. List items are in format vendorID/deviceID and separated by commas. GPU inspection will use this value to count the number of GPU device in a node. If this option is not defined, then leave out pci_gpu_devices in capabilities property. Sample gpu_ids value: 0x1000/0x0079,0x2100/0x0080

fpga_ids

Type list

Default []

List of vendor IDs and device IDs for CPU FPGA to inspect. List items are in format vendorID/deviceID and separated by commas. CPU inspection will use this value to find existence of CPU FPGA in a node. If this option is not defined, then leave out CUSTOM_CPU_FPGA in node traits. Sample fpga_ids value: 0x1000/0x0079,0x2100/0x0080

query_raid_config_fgi_status_interval

Type integer

Default 300

Minimum Value 1

Interval (in seconds) between periodic RAID status checks to determine whether the asynchronous RAID configuration was successfully finished or not. Foreground Initialization (FGI) will start 5 minutes after creating virtual drives.

kernel_append_params

Type string

Default <None>

Mutable This option can be changed without restarting.

Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading /proc/cmdline. Mind severe cmdline size limit! Can be overridden by *instance_info/kernel_append_params* property.

ironic_lib

fatal_exception_format_errors

Type boolean

Default False

Used if there is a formatting error when generating an exception message (a programming error). If True, raise an exception; if False, use the unformatted message.

Table 28: Deprecated Variations

Group	Name
DEFAULT	fatal_exception_format_errors

root_helper

Type string

Default sudo ironic-rootwrap /etc/ironic/rootwrap.conf

Command that is prefixed to commands that are run as root. If not specified, no commands are run as root.

json_rpc

auth_strategy

Type string

Default <None>

Valid Values noauth, keystone, http_basic

Authentication strategy used by JSON RPC. Defaults to the global auth_strategy setting.

Possible values

noauth no authentication

keystone use the Identity service for authentication

http_basic HTTP basic authentication

http_basic_auth_user_file

Type string

Default /etc/ironic/htpasswd-json-rpc

Path to Apache format user authentication file used when auth_strategy=http_basic

host_ip

Type host address

Default ::

The IP address or hostname on which JSON RPC will listen.

port

Type port number

Default 8089

Minimum Value 0

Maximum Value 65535

The port to use for JSON RPC

use_ssl

Type boolean

Default False

Whether to use TLS for JSON RPC

http_basic_username

Type string

Default <None>

Name of the user to use for HTTP Basic authentication client requests.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Use username instead

http_basic_password

Type string

Default <None>

Password to use for HTTP Basic authentication client requests.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Use password instead

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 29: Deprecated Variations

Group	Name
json_rpc	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 30: Deprecated Variations

Group	Name
json_rpc	tenant-id
json_rpc	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 31: Deprecated Variations

Group	Name
json_rpc	tenant-name
json_rpc	tenant_name

split_loggers**Type** boolean**Default** False

Log requests to multiple loggers.

system_scope**Type** unknown type**Default** <None>

Scope for system operations

tenant_id**Type** unknown type**Default** <None>

Tenant ID

tenant_name**Type** unknown type**Default** <None>

Tenant Name

timeout**Type** integer**Default** <None>

Timeout value for http requests

trust_id**Type** unknown type**Default** <None>

Trust ID

user_domain_id**Type** unknown type**Default** <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 32: Deprecated Variations

Group	Name
json_rpc	user-name
json_rpc	user_name

keystone_authtoken

www_authenticate_uri

Type string

Default <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If youre using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

Table 33: Deprecated Variations

Group	Name
keystone_authtoken	auth_uri

auth_uri

Type string

Default <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If youre using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of www_authenticate_uri and will be removed in the S release.

Warning: This option is deprecated for removal since Queens. Its value may be silently ignored in the future.

Reason The `auth_uri` option is deprecated in favor of `www_authenticate_uri` and will be removed in the S release.

auth_version

Type string

Default <None>

API version of the Identity API endpoint.

interface

Type string

Default internal

Interface to use for the Identity API endpoint. Valid values are public, internal (default) or admin.

delay_auth_decision

Type boolean

Default False

Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

http_connect_timeout

Type integer

Default <None>

Request timeout value for communicating with Identity API server.

http_request_max_retries

Type integer

Default 3

How many times are we trying to reconnect when communicating with Identity API Server.

cache

Type string

Default <None>

Request environment key where the Swift cache object is stored. When `auth_token` middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the `memcached_servers` option instead.

certfile

Type string

Default <None>

Required if identity server requires client certificate

keyfile

Type string

Default <None>

Required if identity server requires client certificate

cafile

Type string

Default <None>

A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.

insecure

Type boolean

Default False

Verify HTTPS connections.

region_name

Type string

Default <None>

The region in which the identity server can be found.

memcached_servers

Type list

Default <None>

Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.

Table 34: Deprecated Variations

Group	Name
keystone_authtoken	memcache_servers

token_cache_time

Type integer

Default 300

In order to prevent excessive effort spent validating tokens, the middleware caches previously-seen tokens for a configurable duration (in seconds). Set to -1 to disable caching completely.

memcache_security_strategy

Type string

Default None

Valid Values None, MAC, ENCRYPT

(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token

data is encrypted and authenticated in the cache. If the value is not one of these options or empty, `auth_token` will raise an exception on initialization.

memcache_secret_key

Type string

Default <None>

(Optional, mandatory if `memcache_security_strategy` is defined) This string is used for key derivation.

memcache_pool_dead_retry

Type integer

Default 300

(Optional) Number of seconds memcached server is considered dead before it is tried again.

memcache_pool_maxsize

Type integer

Default 10

(Optional) Maximum total number of open connections to every memcached server.

memcache_pool_socket_timeout

Type integer

Default 3

(Optional) Socket timeout in seconds for communicating with a memcached server.

memcache_pool_unused_timeout

Type integer

Default 60

(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

memcache_pool_conn_get_timeout

Type integer

Default 10

(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.

memcache_use_advanced_pool

Type boolean

Default True

(Optional) Use the advanced (eventlet safe) memcached client pool.

include_service_catalog

Type boolean

Default True

(Optional) Indicate whether to set the X-Service-Catalog header. If False, middleware will not ask for service catalog on token validation and will not set the X-Service-Catalog header.

enforce_token_bind

Type string

Default permissive

Used to control the use and type of token binding. Can be set to: disabled to not check token binding. permissive (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. strict like permissive but if the bind type is unknown the token will be rejected. required any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.

service_token_roles

Type list

Default ['service']

A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.

service_token_roles_required

Type boolean

Default False

For backwards compatibility reasons we must let valid service tokens pass that dont pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.

service_type

Type string

Default <None>

The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 35: Deprecated Variations

Group	Name
keystone_authtoken	auth_plugin

auth_section

Type unknown type

Default <None>

Config Section from which to load plugin specific options

mdns

registration_attempts

Type integer

Default 5

Minimum Value 1

Number of attempts to register a service. Currently has to be larger than 1 because of race conditions in the zeroconf library.

lookup_attempts

Type integer

Default 3

Minimum Value 1

Number of attempts to lookup a service.

params

Type unknown type

Default {}

Additional parameters to pass for the registered service.

interfaces

Type list

Default <None>

List of IP addresses of interfaces to use for mDNS. Defaults to all interfaces on the system.

metrics

backend

Type string

Default noop

Valid Values noop, statsd

Backend to use for the metrics system.

prepend_host

Type boolean

Default False

Prepend the hostname to all metric names. The format of metric names is [global_prefix.][host_name.]prefix.metric_name.

prepend_host_reverse

Type boolean

Default True

Split the prepended host value by . and reverse it (to better match the reverse hierarchical form of domain names).

global_prefix

Type string

Default <None>

Prefix all metric names with this value. By default, there is no global prefix. The format of metric names is [global_prefix.][host_name.]prefix.metric_name.

agent_backend

Type string

Default noop

Backend for the agent ramdisk to use for metrics. Default possible backends are noop and statsd.

agent_prepend_host

Type boolean

Default False

Prepend the hostname to all metric names sent by the agent ramdisk. The format of metric names is [global_prefix.][uuid.][host_name.]prefix.metric_name.

agent_prepend_uuid

Type boolean

Default False

Prepend the nodes Ironic uuid to all metric names sent by the agent ramdisk. The format of metric names is [global_prefix.][uuid.][host_name.]prefix.metric_name.

agent_prepend_host_reverse

Type boolean

Default True

Split the prepended host value by . and reverse it for metrics sent by the agent ramdisk (to better match the reverse hierarchical form of domain names).

agent_global_prefix

Type string

Default <None>

Prefix all metric names sent by the agent ramdisk with this value. The format of metric names is [global_prefix.][uuid.][host_name.]prefix.metric_name.

metrics_statsd

statsd_host

Type string

Default localhost

Host for use with the statsd backend.

statsd_port

Type port number

Default 8125

Minimum Value 0

Maximum Value 65535

Port to use with the statsd backend.

agent_statsd_host

Type string

Default localhost

Host for the agent ramdisk to use with the statsd backend. This must be accessible from networks the agent is booted on.

agent_statsd_port

Type port number

Default 8125

Minimum Value 0

Maximum Value 65535

Port for the agent ramdisk to use with the statsd backend.

molds

storage

Type string

Default swift

Configuration mold storage location. Supports swift and http. By default swift.

user

Type string

Default <None>

User for http Basic auth. By default set empty.

password

Type string

Default <None>

Password for http Basic auth. By default set empty.

retry_attempts

Type string

Default 3

Retry attempts for saving or getting configuration molds.

retry_interval

Type string

Default 3

Retry interval for saving or getting configuration molds.

neutron

add_all_ports

Type boolean

Default False

Mutable This option can be changed without restarting.

Option to enable transmission of all ports to neutron when creating ports for provisioning, cleaning, or rescue. This is done without IP addresses assigned to the port, and may be useful in some bonded network configurations.

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 36: Deprecated Variations

Group	Name
neutron	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile**Type** string**Default** <None>

PEM encoded client certificate cert file

cleaning_network**Type** string**Default** <None>**Mutable** This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for cleaning nodes. Required for neutron network interface. It is also required if cleaning nodes when using flat network interface or neutron DHCP provider. If a name is provided, it must be unique among all networks or cleaning will fail.

Table 37: Deprecated Variations

Group	Name
neutron	cleaning_network_uuid

cleaning_network_security_groups**Type** list**Default** []**Mutable** This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during cleaning of the nodes. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, default security group is used.

collect_timing**Type** boolean**Default** False

Collect per-API call timing information.

connect_retries**Type** integer**Default** <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay**Type** floating point**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

dhcpv6_stateful_address_count

Type integer

Default 4

Mutable This option can be changed without restarting.

Number of IPv6 addresses to allocate for ports created for provisioning, cleaning, rescue or inspection on DHCPv6-stateful networks. Different stages of the chain-loading process will request addresses with different CLID/IAID. Due to non-identical identifiers multiple addresses must be reserved for the host to ensure each step of the boot process can successfully lease addresses.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

inspection_network

Type string

Default <None>

Mutable This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for in-band inspection of nodes. If a name is provided, it must be unique among all networks or inspection will fail.

inspection_network_security_groups

Type list

Default []

Mutable This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during the node inspection process. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, the default security group is used.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

port_setup_delay

Type integer

Default 0

Minimum Value 0

Mutable This option can be changed without restarting.

Delay value to wait for Neutron agents to setup sufficient DHCP configuration for port.

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 38: Deprecated Variations

Group	Name
neutron	tenant-id
neutron	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 39: Deprecated Variations

Group	Name
neutron	tenant-name
neutron	tenant_name

provisioning_network

Type string

Default <None>

Mutable This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for provisioning nodes. Required for neutron network interface. If a name is provided, it must be unique among all networks or deploy will fail.

Table 40: Deprecated Variations

Group	Name
neutron	provisioning_network_uuid

provisioning_network_security_groups**Type** list**Default** []**Mutable** This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during provisioning of the nodes. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, default security group is used.

region_name**Type** string**Default** <None>

The default region_name for endpoint URL discovery.

request_timeout**Type** integer**Default** 45**Mutable** This option can be changed without restarting.

Timeout for request processing when interacting with Neutron. This value should be increased if neutron port action timeouts are observed as neutron performs pre-commit validation prior returning to the API client which can take longer than normal client/server interactions.

rescuing_network**Type** string**Default** <None>**Mutable** This option can be changed without restarting.

Neutron network UUID or name for booting the ramdisk for rescue mode. This is not the network that the rescue ramdisk will use post-boot the tenant network is used for that. Required for neutron network interface, if rescue mode will be used. It is not used for the flat or noop network interfaces. If a name is provided, it must be unique among all networks or rescue will fail.

rescuing_network_security_groups**Type** list**Default** []**Mutable** This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during the node rescue process. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, the default security group is used.

retries**Type** integer**Default** 3**Mutable** This option can be changed without restarting.

DEPRECATED: Client retries in the case of a failed request.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Replaced by `status_code_retries` and `status_code_retry_delay`.

service_name

Type string

Default <None>

The default `service_name` for endpoint URL discovery.

service_type

Type string

Default network

The default `service_type` for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name**Type** unknown type**Default** <None>

Tenant Name

timeout**Type** integer**Default** <None>

Timeout value for http requests

trust_id**Type** unknown type**Default** <None>

Trust ID

user_domain_id**Type** unknown type**Default** <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type**Default** <None>

User id

username**Type** unknown type**Default** <None>

Username

Table 41: Deprecated Variations

Group	Name
neutron	user-name
neutron	user_name

valid_interfaces**Type** list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

nova

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 42: Deprecated Variations

Group	Name
nova	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 43: Deprecated Variations

Group	Name
nova	tenant-id
nova	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 44: Deprecated Variations

Group	Name
nova	tenant-name
nova	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

send_power_notifications

Type boolean

Default True

Mutable This option can be changed without restarting.

When set to True, it will enable the support for power state change callbacks to nova. This option should be set to False in deployments that do not have the openstack compute service.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default compute

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username**Type** unknown type**Default** <None>

Username

Table 45: Deprecated Variations

Group	Name
nova	user-name
nova	user_name

valid_interfaces**Type** list**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version**Type** string**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

oslo_concurrency**disable_process_locking****Type** boolean**Default** False

Enables or disables inter-process locks.

Table 46: Deprecated Variations

Group	Name
DEFAULT	disable_process_locking

lock_path**Type** string**Default** <None>

Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.

Table 47: Deprecated Variations

Group	Name
DEFAULT	lock_path

oslo_messaging_amqp

container_name

Type string

Default <None>

Name for the AMQP container. must be globally unique. Defaults to a generated UUID

Table 48: Deprecated Variations

Group	Name
amqp1	container_name

idle_timeout

Type integer

Default 0

Timeout for inactive connections (in seconds)

Table 49: Deprecated Variations

Group	Name
amqp1	idle_timeout

trace

Type boolean

Default False

Debug: dump AMQP frames to stdout

Table 50: Deprecated Variations

Group	Name
amqp1	trace

ssl

Type boolean

Default False

Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the systems CA-bundle to verify the servers certificate.

ssl_ca_file

Type string

Default ''

CA certificate PEM file used to verify the servers certificate

Table 51: Deprecated Variations

Group	Name
amqp1	ssl_ca_file

ssl_cert_file**Type** string**Default** ''

Self-identifying certificate PEM file for client authentication

Table 52: Deprecated Variations

Group	Name
amqp1	ssl_cert_file

ssl_key_file**Type** string**Default** ''

Private key PEM file used to sign ssl_cert_file certificate (optional)

Table 53: Deprecated Variations

Group	Name
amqp1	ssl_key_file

ssl_key_password**Type** string**Default** <None>

Password for decrypting ssl_key_file (if encrypted)

Table 54: Deprecated Variations

Group	Name
amqp1	ssl_key_password

ssl_verify_vhost**Type** boolean**Default** False

By default SSL checks that the name in the servers certificate matches the hostname in the transport_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl_verify_vhost to True if the servers SSL certificate uses the virtual host name instead of the DNS name.

sasl_mechanisms**Type** string

Default ''

Space separated list of acceptable SASL mechanisms

Table 55: Deprecated Variations

Group	Name
amqp1	sasl_mechanisms

sasl_config_dir

Type string

Default ''

Path to directory that contains the SASL configuration

Table 56: Deprecated Variations

Group	Name
amqp1	sasl_config_dir

sasl_config_name

Type string

Default ''

Name of configuration file (without .conf suffix)

Table 57: Deprecated Variations

Group	Name
amqp1	sasl_config_name

sasl_default_realm

Type string

Default ''

SASL realm to use if no realm present in username

connection_retry_interval

Type integer

Default 1

Minimum Value 1

Seconds to pause before attempting to re-connect.

connection_retry_backoff

Type integer

Default 2

Minimum Value 0

Increase the `connection_retry_interval` by this many seconds after each unsuccessful failover attempt.

connection_retry_interval_max

Type integer

Default 30

Minimum Value 1

Maximum limit for `connection_retry_interval` + `connection_retry_backoff`

link_retry_delay

Type integer

Default 10

Minimum Value 1

Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.

default_reply_retry

Type integer

Default 0

Minimum Value -1

The maximum number of attempts to re-send a reply message which failed due to a recoverable error.

default_reply_timeout

Type integer

Default 30

Minimum Value 5

The deadline for an rpc reply message delivery.

default_send_timeout

Type integer

Default 30

Minimum Value 5

The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.

default_notify_timeout

Type integer

Default 30

Minimum Value 5

The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.

default_sender_link_timeout

Type integer

Default 600

Minimum Value 1

The duration to schedule a purge of idle sender links. Detach link after expiry.

addressing_mode

Type string

Default dynamic

Indicates the addressing mode used by the driver. Permitted values: legacy - use legacy non-routable addressing routable - use routable addresses dynamic - use legacy addresses if the message bus does not support routing otherwise use routable addressing

pseudo_vhost

Type boolean

Default True

Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpid). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private subnet per virtual host. Set to False if the message bus supports virtual hosting using the hostname field in the AMQP 1.0 Open performative as the name of the virtual host.

server_request_prefix

Type string

Default exclusive

address prefix used when sending to a specific server

Table 58: Deprecated Variations

Group	Name
amqp1	server_request_prefix

broadcast_prefix

Type string

Default broadcast

address prefix used when broadcasting to all servers

Table 59: Deprecated Variations

Group	Name
amqp1	broadcast_prefix

group_request_prefix

Type string

Default unicast

address prefix when sending to any server in group

Table 60: Deprecated Variations

Group	Name
amqp1	group_request_prefix

rpc_address_prefix

Type string

Default openstack.org/om/rpc

Address prefix for all generated RPC addresses

notify_address_prefix

Type string

Default openstack.org/om/notify

Address prefix for all generated Notification addresses

multicast_address

Type string

Default multicast

Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.

unicast_address

Type string

Default unicast

Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

anycast_address

Type string

Default anycast

Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.

default_notification_exchange

Type string

Default <None>

Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else notify

default_rpc_exchange

Type string

Default <None>

Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else rpc

reply_link_credit

Type integer

Default 200

Minimum Value 1

Window size for incoming RPC Reply messages.

rpc_server_credit

Type integer

Default 100

Minimum Value 1

Window size for incoming RPC Request messages

notify_server_credit

Type integer

Default 100

Minimum Value 1

Window size for incoming Notification messages

pre_settled

Type multi-valued

Default rpc-cast

Default rpc-reply

Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: rpc-call - send RPC Calls pre-settled rpc-reply- send RPC Replies pre-settled rpc-cast - Send RPC Casts pre-settled notify - Send Notifications pre-settled

oslo_messaging_kafka

kafka_max_fetch_bytes

Type integer

Default 1048576

Max fetch bytes of Kafka consumer

kafka_consumer_timeout

Type floating point

Default 1.0

Default timeout(s) for Kafka consumers

pool_size

Type integer

Default 10

Pool Size for Kafka Consumers

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

conn_pool_min_size

Type integer

Default 2

The pool size limit for connections expiration policy

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

conn_pool_ttl

Type integer

Default 1200

The time-to-live in sec of idle connections in the pool

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

consumer_group

Type string

Default oslo_messaging_consumer

Group id for Kafka consumer. Consumers in one group will coordinate message consumption

producer_batch_timeout

Type floating point

Default 0.0

Upper bound on the delay for KafkaProducer batching in seconds

producer_batch_size

Type integer

Default 16384

Size of batch for the producer async send

compression_codec

Type string

Default none

Valid Values none, gzip, snappy, lz4, zstd

The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version

enable_auto_commit

Type boolean

Default False

Enable asynchronous consumer commits

max_poll_records

Type integer

Default 500

The maximum number of records returned in a poll call

security_protocol

Type string

Default PLAINTEXT

Valid Values PLAINTEXT, SASL_PLAINTEXT, SSL, SASL_SSL

Protocol used to communicate with brokers

sasl_mechanism

Type string

Default PLAIN

Mechanism when security protocol is SASL

ssl_cafile

Type string

Default ''

CA certificate PEM file used to verify the server certificate

ssl_client_cert_file

Type string

Default ''

Client certificate PEM file used for authentication.

ssl_client_key_file

Type string

Default ''

Client key PEM file used for authentication.

ssl_client_key_password

Type string

Default ''

Client key password file used for authentication.

oslo_messaging_notifications

driver

Type multi-valued

Default ''

The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop

Table 61: Deprecated Variations

Group	Name
DEFAULT	notification_driver

transport_url

Type string

Default <None>

A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

Table 62: Deprecated Variations

Group	Name
DEFAULT	notification_transport_url

topics

Type list

Default ['notifications']

AMQP topic used for OpenStack notifications.

Table 63: Deprecated Variations

Group	Name
rpc_notifier2	topics
DEFAULT	notification_topics

retry

Type integer

Default -1

The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite

oslo_messaging_rabbit

amqp_durable_queues

Type boolean

Default False

Use durable queues in AMQP.

amqp_auto_delete

Type boolean

Default False

Auto-delete queues in AMQP.

Table 64: Deprecated Variations

Group	Name
DEFAULT	amqp_auto_delete

ssl

Type boolean

Default False

Connect over SSL.

Table 65: Deprecated Variations

Group	Name
oslo_messaging_rabbit	rabbit_use_ssl

ssl_version

Type string

Default ''

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.

Table 66: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_version

ssl_key_file

Type string

Default ''

SSL key file (valid only if SSL enabled).

Table 67: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_keyfile

ssl_cert_file

Type string

Default ''

SSL cert file (valid only if SSL enabled).

Table 68: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_certfile

ssl_ca_file

Type string

Default ''

SSL certification authority file (valid only if SSL enabled).

Table 69: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_ca_certs

heartbeat_in_pthread

Type boolean

Default True

Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

kombu_reconnect_delay

Type floating point

Default 1.0

How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Table 70: Deprecated Variations

Group	Name
DEFAULT	kombu_reconnect_delay

kombu_compression

Type string

Default <None>

EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.

kombu_missing_consumer_retry_timeout

Type integer

Default 60

How long to wait a missing client before abandoning to send it its replies. This value should not be longer than `rpc_response_timeout`.

Table 71: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_reconnect_timeout

kombu_failover_strategy

Type string

Default round-robin

Valid Values round-robin, shuffle

Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.

rabbit_login_method

Type string

Default AMQPLAIN

Valid Values PLAIN, AMQPLAIN, RABBIT-CR-DEMO

The RabbitMQ login method.

Table 72: Deprecated Variations

Group	Name
DEFAULT	rabbit_login_method

rabbit_retry_interval

Type integer

Default 1

How frequently to retry connecting with RabbitMQ.

rabbit_retry_backoff**Type** integer**Default** 2

How long to backoff for between retries when connecting to RabbitMQ.

Table 73: Deprecated Variations

Group	Name
DEFAULT	rabbit_retry_backoff

rabbit_interval_max**Type** integer**Default** 30

Maximum interval of RabbitMQ connection retries. Default is 30 seconds.

rabbit_ha_queues**Type** boolean**Default** False

Try to use HA queues in RabbitMQ (`x-ha-policy: all`). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the `x-ha-policy` argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: `rabbitmqctl set_policy HA ^(?!amq.).* {ha-mode: all}`

Table 74: Deprecated Variations

Group	Name
DEFAULT	rabbit_ha_queues

rabbit_transient_queues_ttl**Type** integer**Default** 1800**Minimum Value** 1

Positive integer representing duration in seconds for queue TTL (`x-expires`). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.

rabbit_qos_prefetch_count**Type** integer**Default** 0

Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.

heartbeat_timeout_threshold**Type** integer

Default 60

Number of seconds after which the Rabbit broker is considered down if heartbeats keep-alive fails (0 disables heartbeat).

heartbeat_rate

Type integer

Default 2

How often times during the heartbeat_timeout_threshold we check the heartbeat.

direct_mandatory_flag

Type boolean

Default True

(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to lets a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Mandatory flag no longer deactivable.

enable_cancel_on_failover

Type boolean

Default False

Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down

oslo_middleware

enable_proxy_headers_parsing

Type boolean

Default False

Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

oslo_policy

enforce_scope

Type boolean

Default False

This option controls whether or not to enforce scope when evaluating policies. If `True`, the scope of the token used in the request is compared to the `scope_types` of the policy being enforced. If the scopes do not match, an `InvalidScope` exception will be raised. If `False`, a message will be logged informing operators that policies are being invoked with mismatching scope.

enforce_new_defaults

Type boolean

Default False

This option controls whether or not to use old deprecated defaults when evaluating policies. If `True`, the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the `enforce_scope` flag so that you can get the benefits of new defaults and `scope_type` together

policy_file

Type string

Default `policy.yaml`

The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.

Table 75: Deprecated Variations

Group	Name
DEFAULT	<code>policy_file</code>

policy_default_rule

Type string

Default `default`

Default rule. Enforced when a requested rule is not found.

Table 76: Deprecated Variations

Group	Name
DEFAULT	<code>policy_default_rule</code>

policy_dirs

Type multi-valued

Default `policy.d`

Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the `config_dir` option, or absolute paths. The file defined by `policy_file` must exist for these directories to be searched. Missing or empty directories are ignored.

Table 77: Deprecated Variations

Group	Name
DEFAULT	policy_dirs

remote_content_type

Type string

Default application/x-www-form-urlencoded

Valid Values application/x-www-form-urlencoded, application/json

Content Type to send and receive data for REST based policy check

remote_ssl_verify_server_cert

Type boolean

Default False

server identity verification for REST based policy check

remote_ssl_ca_cert_file

Type string

Default <None>

Absolute path to ca cert file for REST based policy check

remote_ssl_client_cert_file

Type string

Default <None>

Absolute path to client cert for REST based policy check

remote_ssl_client_key_file

Type string

Default <None>

Absolute path client key file REST based policy check

profiler

enabled

Type boolean

Default False

Enable the profiling for all services on this node.

Default value is False (fully disable the profiling feature).

Possible values:

- True: Enables the feature
- False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.

Table 78: Deprecated Variations

Group	Name
profiler	profiler_enabled

trace_sqlalchemy**Type** boolean**Default** False

Enable SQL requests profiling in services.

Default value is False (SQL requests wont be traced).

Possible values:

- True: Enables SQL requests profiling. Each SQL query will be part of the trace and can the be analyzed by how much time was spent for that.
- False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

hmac_keys**Type** string**Default** SECRET_KEY

Secret key(s) to use for encrypting context data for performance profiling.

This string value should have the following format: <key1>[,<key2>,<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project.

Both enabled flag and hmac_keys config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.

connection_string**Type** string**Default** messaging://

Connection string for a notifier backend.

Default value is `messaging://` which sets the notifier to `oslo_messaging`.

Examples of possible values:

- `messaging://` - use `oslo_messaging` driver for sending spans.
- `redis://127.0.0.1:6379` - use `redis` driver for sending spans.
- `mongodb://127.0.0.1:27017` - use `mongodb` driver for sending spans.

- `elasticsearch://127.0.0.1:9200` - use elasticsearch driver for sending spans.
- `jaeger://127.0.0.1:6831` - use jaeger tracing as driver for sending spans.

es_doc_type

Type string

Default notification

Document type for notification indexing in elasticsearch.

es_scroll_time

Type string

Default 2m

This parameter is a time value parameter (for example: `es_scroll_time=2m`), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

es_scroll_size

Type integer

Default 10000

Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: `es_scroll_size=10000`).

socket_timeout

Type floating point

Default 0.1

Redissentinel provides a timeout option on the connections. This parameter defines that timeout (for example: `socket_timeout=0.1`).

sentinel_service_name

Type string

Default mymaster

Redissentinel uses a service name to identify a master redis service. This parameter defines the name (for example: `sentinal_service_name=mymaster`).

filter_error_trace

Type boolean

Default False

Enable filter traces that contain error/exception to a separated place.

Default value is set to False.

Possible values:

- True: Enable filter traces that contain error/exception.
- False: Disable the filter.

pxe**kernel_append_params****Type** string**Default** nofb nomodeset vga=normal**Mutable** This option can be changed without restarting.

Additional append parameters for baremetal PXE boot.

Table 79: Deprecated Variations

Group	Name
pxe	pxe_append_params

default_ephemeral_format**Type** string**Default** ext4**Mutable** This option can be changed without restarting.

Default file system format for ephemeral partition, if one is created.

images_path**Type** string**Default** /var/lib/ironic/images/

On the ironic-conductor node, directory where images are stored on disk.

instance_master_path**Type** string**Default** /var/lib/ironic/master_images

On the ironic-conductor node, directory where master instance images are stored on disk. Setting to the empty string disables image caching.

image_cache_size**Type** integer**Default** 20480

Maximum size (in MiB) of cache for master images, including those in use.

image_cache_ttl**Type** integer**Default** 10080

Maximum TTL (in minutes) for old master images in cache.

pxe_config_template**Type** string**Default** \$pybasedir/drivers/modules/pxe_config.template

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE loader configuration.

ipxe_config_template

Type string

Default `$pybasedir/drivers/modules/ipxe_config.template`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for iPXE operations.

uefi_pxe_config_template

Type string

Default `$pybasedir/drivers/modules/pxe_grub_config.template`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE configuration for UEFI boot loader. Generally this is used for GRUB specific templates.

pxe_config_template_by_arch

Type dict

Default `{}`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE configuration per node architecture. For example: `aarch64:/opt/share/grubaa64_pxe_config.template`

tftp_server

Type string

Default `$my_ip`

IP address of ironic-conductor nodes TFTP server.

tftp_root

Type string

Default `/tftpboot`

ironic-conductor nodes TFTP root path. The ironic-conductor must have read/write access to this path.

tftp_master_path

Type string

Default `/tftpboot/master_images`

On ironic-conductor node, directory where master TFTP images are stored on disk. Setting to the empty string disables image caching.

dir_permission

Type integer

Default `<None>`

The permission that will be applied to the TFTP folders upon creation. This should be set to the permission such that the tftpserver has access to read the contents of the configured TFTP folder. This setting is only required when the operating systems umask is restrictive such that ironic-conductor is creating files that cannot be read by the TFTP server. Setting to <None> will result in the operating systems umask to be utilized for the creation of new tftp folders. It is recommended that an octal representation is specified. For example: 0o755

pxe_bootfile_name**Type** string**Default** pxelinux.0

Bootfile DHCP parameter.

pxe_config_subdir**Type** string**Default** pxelinux.cfg

Directory in which to create symbolic links which represent the MAC or IP address of the ports on a node and allow boot loaders to load the PXE file for the node. This directory name is relative to the PXE or iPXE folders.

uefi_pxe_bootfile_name**Type** string**Default** bootx64.efi

Bootfile DHCP parameter for UEFI boot mode.

ipxe_bootfile_name**Type** string**Default** undionly.kpxe

Bootfile DHCP parameter.

uefi_ipxe_bootfile_name**Type** string**Default** ipxe.efi

Bootfile DHCP parameter for UEFI boot mode. If you experience problems with booting using it, try snponly.efi.

pxe_bootfile_name_by_arch**Type** dict**Default** {}

Bootfile DHCP parameter per node architecture. For example: aarch64:grubaa64.efi

ipxe_bootfile_name_by_arch**Type** dict**Default** {}

Bootfile DHCP parameter per node architecture. For example: aarch64:ipxe_aa64.efi

ipxe_boot_script

Type string

Default \$pybasedir/drivers/modules/boot.ipxe

On ironic-conductor node, the path to the main iPXE script file.

ipxe_timeout

Type integer

Default 0

Timeout value (in seconds) for downloading an image via iPXE. Defaults to 0 (no timeout)

boot_retry_timeout

Type integer

Default <None>

Minimum Value 60

Timeout (in seconds) after which PXE boot should be retried. Must be less than [conductor]deploy_callback_timeout. Disabled by default.

boot_retry_check_interval

Type integer

Default 90

Minimum Value 1

Interval (in seconds) between periodic checks on PXE boot retry. Has no effect if boot_retry_timeout is not set.

ip_version

Type string

Default 4

Valid Values 4, 6

Mutable This option can be changed without restarting.

The IP version that will be used for PXE booting. Defaults to 4. EXPERIMENTAL

Possible values

4 IPv4

6 IPv6

ipxe_use_swift

Type boolean

Default False

Mutable This option can be changed without restarting.

Download deploy and rescue images directly from swift using temporary URLs. If set to false (default), images are downloaded to the ironic-conductor node and served over its local HTTP server. Applicable only when ipxe compatible boot interface is used.

enable_netboot_fallback

Type boolean

Default False

Mutable This option can be changed without restarting.

If True, generate a PXE environment even for nodes that use local boot. This is useful when the driver cannot switch nodes to local boot, e.g. with SNMP or with Redfish on machines that cannot do persistent boot. Mostly useful for standalone ironic since Neutron will prevent incorrect PXE boot.

redfish**connection_attempts**

Type integer

Default 5

Minimum Value 1

Maximum number of attempts to try to connect to Redfish

connection_retry_interval

Type integer

Default 4

Minimum Value 1

Number of seconds to wait between attempts to connect to Redfish

connection_cache_size

Type integer

Default 1000

Minimum Value 0

Maximum Redfish client connection cache size. Redfish driver would strive to reuse authenticated BMC connections (obtained through Redfish Session Service). This option caps the maximum number of connections to maintain. The value of 0 disables client connection caching completely.

auth_type

Type string

Default auto

Valid Values basic, session, auto

Redfish HTTP client authentication method.

Possible values

basic Use HTTP basic authentication

session Use HTTP session authentication

auto Try HTTP session authentication first, fall back to basic HTTP authentication

use_swift

Type boolean

Default True

Mutable This option can be changed without restarting.

Upload generated ISO images for virtual media boot to Swift, then pass temporary URL to BMC for booting the node. If set to false, images are placed on the ironic-conductor node and served over its local HTTP server.

swift_container

Type string

Default ironic_redfish_container

Mutable This option can be changed without restarting.

The Swift container to store Redfish driver data. Applies only when *use_swift* is enabled.

swift_object_expiry_timeout

Type integer

Default 900

Mutable This option can be changed without restarting.

Amount of time in seconds for Swift objects to auto-expire. Applies only when *use_swift* is enabled.

kernel_append_params

Type string

Default nofb nomodeset vga=normal

Mutable This option can be changed without restarting.

Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading `/proc/cmdline`. Mind severe cmdline size limit! Can be overridden by *instance_info/kernel_append_params* property.

file_permission

Type integer

Default 420

File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644, or as the octal number `0o644` in Python. This setting must be set to the octal number representation, meaning starting with `0o`.

firmware_update_status_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for completed firmware update tasks

firmware_update_fail_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for failed firmware update tasks

raid_config_status_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for completed raid config tasks

raid_config_fail_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for failed raid config tasks

service_catalog

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 80: Deprecated Variations

Group	Name
service_catalog	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with `min_version`. Mutually exclusive with `version`.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with `max_version`. Mutually exclusive with `version`. If `min_version` is given with no `max_version` it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 81: Deprecated Variations

Group	Name
service_catalog	tenant-id
service_catalog	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 82: Deprecated Variations

Group	Name
service_catalog	tenant-name
service_catalog	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default baremetal

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 83: Deprecated Variations

Group	Name
service_catalog	user-name
service_catalog	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

snmp

power_timeout

Type integer

Default 10

Seconds to wait for power action to be completed

reboot_delay

Type integer

Default 0

Minimum Value 0

Time (in seconds) to sleep between when rebooting (powering off and on again)

udp_transport_timeout

Type floating point

Default 1.0

Minimum Value 0.0

Response timeout in seconds used for UDP transport. Timeout should be a multiple of 0.5 seconds and is applicable to each retry.

udp_transport_retries

Type integer

Default 5

Minimum Value 0

Maximum number of UDP request retries, 0 means no retries.

ssl

ca_file

Type string

Default <None>

CA certificate file to use to verify connecting clients.

Table 84: Deprecated Variations

Group	Name
DEFAULT	ssl_ca_file

cert_file

Type string

Default <None>

Certificate file to use when starting the server securely.

Table 85: Deprecated Variations

Group	Name
DEFAULT	ssl_cert_file

key_file

Type string

Default <None>

Private key file to use when starting the server securely.

Table 86: Deprecated Variations

Group	Name
DEFAULT	ssl_key_file

version**Type** string**Default** <None>

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.

ciphers**Type** string**Default** <None>

Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.

swift**auth_url****Type** unknown type**Default** <None>

Authentication URL

auth_type**Type** unknown type**Default** <None>

Authentication type to load

Table 87: Deprecated Variations

Group	Name
swift	auth_plugin

cafile**Type** string**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile**Type** string**Default** <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 88: Deprecated Variations

Group	Name
swift	tenant-id
swift	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 89: Deprecated Variations

Group	Name
swift	tenant-name
swift	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default object-store

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

swift_max_retries

Type integer

Default 2

Maximum number of times to retry a Swift request, before failing.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 90: Deprecated Variations

Group	Name
swift	user-name
swift	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

xclarity

manager_ip

Type string

Default <None>

IP address of the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_manager_ip instead

username

Type string

Default <None>

Username for the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_username instead

password

Type string

Default <None>

Password for XClarity Controller username. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_password instead

port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Port to be used for XClarity Controller connection.

5.1.2 Policies

Warning: JSON formatted policy files were deprecated in the Wallaby development cycle due to the Victoria deprecation by the `olso.policy` library. Use the `oslopolicy-convert-json-to-yaml` tool to convert the existing JSON to YAML formatted policy file in backward compatible way.

The following is an overview of all available policies in Ironic. For a sample configuration file, refer to *Ironic Policy*.

ironic.api

admin_api

Default role:admin or role:administrator

Legacy rule for cloud admin access

public_api

Default is_public_api:True

Internal flag for public API routes

show_password

Default !

Show or mask secrets within node driver information in API responses

show_instance_secrets

Default !

Show or mask secrets within instance information in API responses

is_member

Default (project_domain_id:default or project_domain_id:None) and
(project_name:demo or project_name:baremetal)

May be used to restrict access to specific projects

is_observer

Default rule:is_member and (role:observer or role:baremetal_observer)

Read-only API access

is_admin

Default rule:admin_api or (rule:is_member and role:baremetal_admin)

Full read/write API access

is_node_owner

Default project_id:%(node.owner)s

Owner of node

is_node_lessee

Default project_id:%(node.lessee)s

Lessee of node

is_allocation_owner

Default project_id:%(allocation.owner)s

Owner of allocation

baremetal:node:create

Default role:admin and system_scope:all

Operations

- **POST** /nodes

Scope Types

- **system**

Create Node records

baremetal:node:list

Default role:reader

Operations

- **GET** /nodes
- **GET** /nodes/detail

Scope Types

- **system**
- **project**

Retrieve multiple Node records, filtered by an explicit owner or the client project_id

baremetal:node:list_all

Default role:reader and system_scope:all

Operations

- GET /nodes
- GET /nodes/detail

Scope Types

- system

Retrieve multiple Node records

baremetal:node:get

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s or project_id:%(node.lessee)s)

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Retrieve a single Node record

baremetal:node:get:filter_threshold

Default role:reader and system_scope:all

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Filter to allow operators to govern the threshold where information should be filtered. Non-authorized users will be subjected to additional API policy checks for API content response bodies.

baremetal:node:get:last_error

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Governs if the node last_error field is masked from APIclients with insufficient privileges.

baremetal:node:get:reservation

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Governs if the node reservation field is masked from APIclients with insufficient privileges.

baremetal:node:get:driver_internal_info

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Governs if the node driver_internal_info field is masked from API clients with insufficient privileges.

baremetal:node:get:driver_info

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- GET /nodes/{node_ident}

Scope Types

- system
- project

Governs if the driver_info field is masked from APIclients with insufficient privileges.

baremetal:node:update:driver_info

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- PATCH /nodes/{node_ident}

Scope Types

- system
- project

Governs if node driver_info field can be updated via the API clients.

baremetal:node:update:properties

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node properties field can be updated via the API clients.

baremetal:node:update:chassis_uuid

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node chassis_uuid field can be updated via the API clients.

baremetal:node:update:instance_uuid

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node instance_uuid field can be updated via the API clients.

baremetal:node:update:lessee

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node lessee field can be updated via the API clients.

baremetal:node:update:owner

Default role:member and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node owner field can be updated via the API clients.

baremetal:node:update:driver_interfaces

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node driver and driver interfaces field can be updated via the API clients.

baremetal:node:update:network_data

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node driver_info field can be updated via the API clients.

baremetal:node:update:conductor_group

Default role:member and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node conductor_group field can be updated via the API clients.

baremetal:node:update:name

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node name field can be updated via the API clients.

baremetal:node:update:retired

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node retired and retired reason can be updated by API clients.

baremetal:node:update

Default (role:member and system_scope:all) or (role:member and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Generalized update of node records

baremetal:node:update_extra

Default (role:member and system_scope:all) or (role:member and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Update Node extra field

baremetal:node:update_instance_info

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Update Node instance_info field

baremetal:node:update_owner_provisioned

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**

Update Node owner even when Node is provisioned

baremetal:node:delete

Default role:admin and system_scope:all

Operations

- **DELETE** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Delete Node records

baremetal:node:validate

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **GET** /nodes/{node_ident}/validate

Scope Types

- **system**
- **project**

Request active validation of Nodes

baremetal:node:set_maintenance

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PUT** /nodes/{node_ident}/maintenance

Scope Types

- **system**
- **project**

Set maintenance flag, taking a Node out of service

baremetal:node:clear_maintenance

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **DELETE** /nodes/{node_ident}/maintenance

Scope Types

- **system**
- **project**

Clear maintenance flag, placing the Node into service again

baremetal:node:get_boot_device

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}/management/boot_device
- **GET** /nodes/{node_ident}/management/boot_device/supported

Scope Types

- **system**
- **project**

Retrieve Node boot device metadata

baremetal:node:set_boot_device

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/boot_device

Scope Types

- **system**

- **project**

Change Node boot device

baremetal:node:get_indicator_state

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/management/indicators/{component}/{indicator}
- **GET** /nodes/{node_ident}/management/indicators

Scope Types

- **system**
- **project**

Retrieve Node indicators and their states

baremetal:node:set_indicator_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/indicators/{component}/{indicator}

Scope Types

- **system**
- **project**

Change Node indicator state

baremetal:node:inject_nmi

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/inject_nmi

Scope Types

- **system**
- **project**

Inject NMI for a node

baremetal:node:get_states

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/states

Scope Types

- **system**
- **project**

View Node power and provision state

baremetal:node:set_power_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s or project_id:%(node.lessee)s)

Operations

- **PUT** /nodes/{node_ident}/states/power

Scope Types

- **system**
- **project**

Change Node power status

baremetal:node:set_provision_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PUT** /nodes/{node_ident}/states/provision

Scope Types

- **system**
- **project**

Change Node provision status

baremetal:node:set_raid_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/states/raid

Scope Types

- **system**
- **project**

Change Node RAID status

baremetal:node:get_console

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}/states/console

Scope Types

- **system**
- **project**

Get Node console connection information

baremetal:node:set_console_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/states/console

Scope Types

- **system**
- **project**

Change Node console status

baremetal:node:vif:list

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/vifs

Scope Types

- **system**
- **project**

List VIFs attached to node

baremetal:node:vif:attach

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **POST** /nodes/{node_ident}/vifs

Scope Types

- **system**
- **project**

Attach a VIF to a node

baremetal:node:vif:detach

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **DELETE** /nodes/{node_ident}/vifs/{node_vif_ident}

Scope Types

- **system**
- **project**

Detach a VIF from a node

baremetal:node:traits:list

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/traits

Scope Types

- **system**
- **project**

List node traits

baremetal:node:traits:set

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/traits
- **PUT** /nodes/{node_ident}/traits/{trait}

Scope Types

- **system**
- **project**

Add a trait to, or replace all traits of, a node

baremetal:node:traits:delete

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /nodes/{node_ident}/traits
- **DELETE** /nodes/{node_ident}/traits/{trait}

Scope Types

- **system**

- **project**

Remove one or all traits from a node

baremetal:node:bios:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/bios
- **GET** /nodes/{node_ident}/bios/{setting}

Scope Types

- **system**
- **project**

Retrieve Node BIOS information

baremetal:node:disable_cleaning

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**

Disable Node disk cleaning

baremetal:port:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /ports/{port_id}
- **GET** /nodes/{node_ident}/ports
- **GET** /nodes/{node_ident}/ports/detail
- **GET** /portgroups/{portgroup_ident}/ports
- **GET** /portgroups/{portgroup_ident}/ports/detail

Scope Types

- **system**
- **project**

Retrieve Port records

baremetal:port:list

Default role:reader

Operations

- **GET** /ports
- **GET** /ports/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records, filtered by owner

baremetal:port:list_all

Default role:reader and system_scope:all

Operations

- **GET** /ports
- **GET** /ports/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records

baremetal:port:create

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **POST** /ports

Scope Types

- **system**
- **project**

Create Port records

baremetal:port:delete

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /ports/{port_id}

Scope Types

- **system**
- **project**

Delete Port records

baremetal:port:update

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /ports/{port_id}

Scope Types

- **system**
- **project**

Update Port records

baremetal:portgroup:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /portgroups
- **GET** /portgroups/detail
- **GET** /portgroups/{portgroup_ident}
- **GET** /nodes/{node_ident}/portgroups
- **GET** /nodes/{node_ident}/portgroups/detail

Scope Types

- **system**
- **project**

Retrieve Portgroup records

baremetal:portgroup:create

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **POST** /portgroups

Scope Types

- **system**
- **project**

Create Portgroup records

baremetal:portgroup:delete

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /portgroups/{portgroup_ident}

Scope Types

- **system**
- **project**

Delete Portgroup records

baremetal:portgroup:update

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /portgroups/{portgroup_ident}

Scope Types

- **system**
- **project**

Update Portgroup records

baremetal:portgroup:list

Default role:reader

Operations

- **GET** /portgroups
- **GET** /portgroups/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records, filtered by owner

baremetal:portgroup:list_all

Default role:reader and system_scope:all

Operations

- **GET** /portgroups
- **GET** /portgroups/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records

baremetal:chassis:get

Default role:reader and system_scope:all

Operations

- **GET** /chassis
- **GET** /chassis/detail

- **GET** /chassis/{chassis_id}

Scope Types

- **system**

Retrieve Chassis records

baremetal:chassis:create

Default role:admin and system_scope:all

Operations

- **POST** /chassis

Scope Types

- **system**

Create Chassis records

baremetal:chassis:delete

Default role:admin and system_scope:all

Operations

- **DELETE** /chassis/{chassis_id}

Scope Types

- **system**

Delete Chassis records

baremetal:chassis:update

Default role:member and system_scope:all

Operations

- **PATCH** /chassis/{chassis_id}

Scope Types

- **system**

Update Chassis records

baremetal:driver:get

Default role:reader and system_scope:all

Operations

- **GET** /drivers
- **GET** /drivers/{driver_name}

Scope Types

- **system**

View list of available drivers

baremetal:driver:get_properties

Default role:reader and system_scope:all

Operations

- **GET** /drivers/{driver_name}/properties

Scope Types

- **system**

View driver-specific properties

baremetal:driver:get_raid_logical_disk_properties

Default role:reader and system_scope:all

Operations

- **GET** /drivers/{driver_name}/raid/logical_disk_properties

Scope Types

- **system**

View driver-specific RAID metadata

baremetal:node:vendor_passthru

Default role:admin and system_scope:all

Operations

- **GET** nodes/{node_ident}/vendor_passthru/methods
- **GET** nodes/{node_ident}/vendor_passthru? method={method_name}
- **PUT** nodes/{node_ident}/vendor_passthru? method={method_name}
- **POST** nodes/{node_ident}/vendor_passthru? method={method_name}
- **PATCH** nodes/{node_ident}/vendor_passthru? method={method_name}
- **DELETE** nodes/{node_ident}/vendor_passthru? method={method_name}

Scope Types

- **system**
- **project**

Access vendor-specific Node functions

baremetal:driver:vendor_passthru

Default role:admin and system_scope:all

Operations

- **GET** drivers/{driver_name}/vendor_passthru/methods

- **GET** `drivers/{driver_name}/vendor_passthru?method={method_name}`
- **PUT** `drivers/{driver_name}/vendor_passthru?method={method_name}`
- **POST** `drivers/{driver_name}/vendor_passthru?method={method_name}`
- **PATCH** `drivers/{driver_name}/vendor_passthru?method={method_name}`
- **DELETE** `drivers/{driver_name}/vendor_passthru?method={method_name}`

Scope Types

- **system**

Access vendor-specific Driver functions

baremetal:node:ipa_heartbeat

Default <empty string>

Operations

- **POST** `/heartbeat/{node_ident}`

Receive heartbeats from IPA ramdisk

baremetal:driver:ipa_lookup

Default <empty string>

Operations

- **GET** `/lookup`

Access IPA ramdisk functions

baremetal:volume:list_all

Default `role:reader and system_scope:all`

Operations

- **GET** `/volume/connectors`
- **GET** `/volume/targets`
- **GET** `/nodes/{node_ident}/volume/connectors`
- **GET** `/nodes/{node_ident}/volume/targets`

Scope Types

- **system**
- **project**

Retrieve a list of all Volume connector and target records

baremetal:volume:list

Default `role:reader`

Operations

- **GET** /volume/connectors
- **GET** /volume/targets
- **GET** /nodes/{node_ident}/volume/connectors
- **GET** /nodes/{node_ident}/volume/targets

Scope Types

- **system**
- **project**

Retrieve a list of Volume connector and target records

baremetal:volume:get

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s or project_id:%(node.lessee)s)

Operations

- **GET** /volume
- **GET** /volume/connectors
- **GET** /volume/connectors/{volume_connector_id}
- **GET** /volume/targets
- **GET** /volume/targets/{volume_target_id}
- **GET** /nodes/{node_ident}/volume
- **GET** /nodes/{node_ident}/volume/connectors
- **GET** /nodes/{node_ident}/volume/targets

Scope Types

- **system**
- **project**

Retrieve Volume connector and target records

baremetal:volume:create

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **POST** /volume/connectors
- **POST** /volume/targets

Scope Types

- **system**
- **project**

Create Volume connector and target records

baremetal:volume:delete

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **DELETE** /volume/connectors/{volume_connector_id}
- **DELETE** /volume/targets/{volume_target_id}

Scope Types

- **system**
- **project**

Delete Volume connector and target records

baremetal:volume:update

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PATCH** /volume/connectors/{volume_connector_id}
- **PATCH** /volume/targets/{volume_target_id}

Scope Types

- **system**
- **project**

Update Volume connector and target records

baremetal:volume:view_target_properties

Default (role:reader and system_scope:all) or (role:admin)

Operations

- **GET** /volume/connectors/{volume_connector_id}
- **GET** /volume/targets/{volume_target_id}

Scope Types

- **system**
- **project**

Ability to view volume target properties

baremetal:conductor:get

Default role:reader and system_scope:all

Operations

- **GET** /conductors

- **GET** /conductors/{hostname}

Scope Types

- **system**

Retrieve Conductor records

baremetal:allocation:get

Default (role:reader and system_scope:all) or (role:reader and project_id:%(allocation.owner)s)

Operations

- **GET** /allocations/{allocation_id}
- **GET** /nodes/{node_ident}/allocation

Scope Types

- **system**
- **project**

Retrieve Allocation records

baremetal:allocation:list

Default role:reader

Operations

- **GET** /allocations

Scope Types

- **system**
- **project**

Retrieve multiple Allocation records, filtered by owner

baremetal:allocation:list_all

Default role:reader and system_scope:all

Operations

- **GET** /allocations

Scope Types

- **system**
- **project**

Retrieve multiple Allocation records

baremetal:allocation:create

Default (role:member and system_scope:all) or (role:member)

Operations

- **POST** /allocations

Scope Types

- **system**
- **project**

Create Allocation records

baremetal:allocation:create_restricted

Default role:member and system_scope:all

Operations

- **POST** /allocations

Scope Types

- **system**
- **project**

Create Allocation records with a specific owner.

baremetal:allocation:delete

Default (role:member and system_scope:all) or (role:member and project_id:%(allocation.owner)s)

Operations

- **DELETE** /allocations/{allocation_id}
- **DELETE** /nodes/{node_ident}/allocation

Scope Types

- **system**
- **project**

Delete Allocation records

baremetal:allocation:update

Default (role:member and system_scope:all) or (role:member and project_id:%(allocation.owner)s)

Operations

- **PATCH** /allocations/{allocation_id}

Scope Types

- **system**
- **project**

Change name and extra fields of an allocation

baremetal:allocation:create_pre_rbac

Default (rule:is_member and role:baremetal_admin) or (is_admin_project:True and role:admin)

Operations

- **PATCH** /allocations/{allocation_id}

Scope Types

- **project**

Logical restrictor to prevent legacy allocation rule missuse - Requires blank allocations to originate from the legacy baremetal_admin.

baremetal:events:post

Default role:admin and system_scope:all

Operations

- **POST** /events

Scope Types

- **system**

Post events

baremetal:deploy_template:get

Default role:reader and system_scope:all

Operations

- **GET** /deploy_templates
- **GET** /deploy_templates/{deploy_template_ident}

Scope Types

- **system**

Retrieve Deploy Template records

baremetal:deploy_template:create

Default role:admin and system_scope:all

Operations

- **POST** /deploy_templates

Scope Types

- **system**

Create Deploy Template records

baremetal:deploy_template:delete

Default role:admin and system_scope:all

Operations

- **DELETE** /deploy_templates/{deploy_template_ident}

Scope Types

- **system**

Delete Deploy Template records

baremetal:deploy_template:update

Default role:admin and system_scope:all

Operations

- **PATCH** /deploy_templates/{deploy_template_ident}

Scope Types

- **system**

Update Deploy Template records

BARE METAL API REFERENCES

Ironics REST API has changed since its first release, and continues to evolve to meet the changing needs of the community. Here we provide a conceptual guide as well as more detailed reference documentation.

6.1 REST API Conceptual Guide

6.1.1 Versioning

The ironic REST API supports two types of versioning:

- major versions, which have dedicated urls.
- microversions, which can be requested through the use of the `X-OpenStack-Ironic-API-Version` header.

There is only one major version supported currently, v1. As such, most URLs in this documentation are written with the `/v1/` prefix.

Starting with the Kilo release, ironic supports microversions. In this context, a version is defined as a string of 2 integers separated by a dot: **X.Y**. Here **X** is a major version, always equal to 1, and **Y** is a minor version. Server minor version is increased every time the API behavior is changed (note *Exceptions from Versioning*).

Note: [Nova versioning documentation](#) has a nice guide for developers on when to bump an API version.

The server indicates its minimum and maximum supported API versions in the `X-OpenStack-Ironic-API-Minimum-Version` and `X-OpenStack-Ironic-API-Maximum-Version` headers respectively, returned with every response. Client may request a specific API version by providing `X-OpenStack-Ironic-API-Version` header with request.

The requested microversion determines both the allowable requests and the response format for all requests. A resource may be represented differently based on the requested microversion.

If no version is requested by the client, the minimum supported version will be assumed. In this way, a client is only exposed to those API features that are supported in the requested (explicitly or implicitly) API version (again note *Exceptions from Versioning*, they are not covered by this rule).

We recommend clients that require a stable API to always request a specific version of API that they have been tested against.

Note: A special value `latest` can be requested instead a numerical microversion, which always requests the newest supported API version from the server.

REST API Versions History

REST API Version History

1.74 (Xena, 18.0)

Add support for BIOS registry fields which include details about the BIOS setting. Included in the `/v1/nodes/{node_ident}/bios/{setting}` response. Add a new selector to include the fields in the BIOS settings list: `* /v1/nodes/{node_ident}/bios/?detail=` Also add a fields selector to the the BIOS settings list: `* /v1/nodes/{node_ident}/bios/?fields=`

1.73 (Xena, 18.0)

Add a new `deploy` verb as an alias to `active` and `undeploy` verb as an alias to `deleted`.

1.72 (Wallaby, 17.0)

Add support for `agent_status` and `agent_status_message` to `/v1/heartbeat`. These fields are used for external installation tools, such as Anaconda, to report back status.

1.71 (Wallaby, 17.0)

Signifier of the API supporting keystone `system` scoped roles and access controls. This is an informational flag for clients to be aware of the servers capability.

1.70 (Wallaby, 17.0)

Add support for `disable_ramdisk` parameter to provisioning endpoint `/v1/nodes/{node_ident}/states/provision`.

1.69 (Wallaby, 16.2)

Add support for `deploy_steps` parameter to provisioning endpoint `/v1/nodes/{node_ident}/states/provision`. Available and optional when target is active or rebuild.

1.68 (Victoria, 16.0)

Added the `agent_verify_ca` parameter to the ramdisk heartbeat API.

1.67 (Victoria, 15.1)

Add support for the mutually exclusive `port_uuid` and `portgroup_uuid` fields by having the `node_vif_attach` API accept those values within `vif_info`. If one is specified, then Ironic will attempt to attach a VIF to the relative port or portgroup.

1.66 (Victoria, 15.1)

Add `network_data` field to the node object, that will be used by stand-alone ironic to pass L3 network configuration information to ramdisk.

1.65 (Ussuri, 15.0)

Added `lessee` field to the node object. The field should match the `project_id` of the intended lessee. If an allocation has an owner, then the allocation process will only match the allocation with a node that has the same owner or lessee.

1.64 (Ussuri, 15.0)

Added the `network_type` to the port objects `local_link_connection` field. The `network_type` can be set to either `managed` or `unmanaged`. When the type is `unmanaged` other fields are not required. Use `unmanaged` when the neutron `network_interface` is required, but the network is in fact a flat network where no actual switch management is done.

1.63 (Ussuri, 15.0)

Added the following new endpoints for indicator management:

- GET `/v1/nodes/<node_ident>/management/indicators` to list all available indicators names for each of the hardware component. Currently known components are: `chassis`, `system`, `disk`, `power` and `nic`.
- GET `/v1/nodes/<node_ident>/management/indicators/<component>/<indicator_ident>` to retrieve all indicators and their states for the hardware component.
- PUT `/v1/nodes/<node_ident>/management/indicators/<component>/<indicator_ident>` change state of the desired indicators of the component.

1.62 (Ussuri, 15.0)

This version of the API is to signify capability of an ironic deployment to support the `agent token` functionality with the `ironic-python-agent`.

1.61 (Ussuri, 14.0)

Added `retired` field to the node object to mark nodes for retirement. If set, this flag will move nodes to `manageable` upon automatic cleaning. `manageable` nodes which have this flag set cannot be moved to `available`. Also added `retired_reason` to specify the retirement reason.

1.60 (Ussuri, 14.0)

Added `owner` field to the allocation object. The field should match the `project_id` of the intended owner. If the `owner` field is set, the allocation process will only match the allocation with a node that has the same `owner` field set.

1.59 (Ussuri, 14.0)

Added the ability to specify a `vendor_data` dictionary field in the `configdrive` parameter submitted with the deployment of a node. The value is a dictionary which is served as `vendor_data2.json` in the config drive.

1.58 (Train, 12.2.0)

Added the ability to backfill allocations for already deployed nodes by creating an allocation with `node set`.

1.57 (Train, 12.2.0)

Added the following new endpoint for allocation:

- `PATCH /v1/allocations/<allocation_ident>` that allows updating name and extra fields for an existing allocation.

1.56 (Stein, 12.1.0)

Added the ability for the `configdrive` parameter submitted with the deployment of a node, to include a `meta_data`, `network_data` and `user_data` dictionary fields. Ironic will now use the supplied data to create a configuration drive for the user. Prior uses of the `configdrive` field are unaffected.

1.55 (Stein, 12.1.0)

Added the following new endpoints for deploy templates:

- GET `/v1/deploy_templates` to list all deploy templates.
- GET `/v1/deploy_templates/<deploy template identifier>` to retrieve details of a deploy template.
- POST `/v1/deploy_templates` to create a deploy template.
- PATCH `/v1/deploy_templates/<deploy template identifier>` to update a deploy template.
- DELETE `/v1/deploy_templates/<deploy template identifier>` to delete a deploy template.

1.54 (Stein, 12.1.0)

Added new endpoints for external events:

- POST `/v1/events` for creating events. (This endpoint is only intended for internal consumption.)

1.53 (Stein, 12.1.0)

Added `is_smartnic` field to the port object to enable Smart NIC port creation in addition to local link connection attributes `port_id` and `hostname`.

1.52 (Stein, 12.1.0)

Added allocation API, allowing reserving a node for deployment based on resource class and traits. The new endpoints are:

- POST `/v1/allocations` to request an allocation.
- GET `/v1/allocations` to list all allocations.
- GET `/v1/allocations/<ID or name>` to retrieve the allocation details.
- GET `/v1/nodes/<ID or name>/allocation` to retrieve an allocation associated with the node.
- DELETE `/v1/allocations/<ID or name>` to remove the allocation.
- DELETE `/v1/nodes/<ID or name>/allocation` to remove an allocation associated with the node.

Also added a new field `allocation_uuid` to the node resource.

1.51 (Stein, 12.1.0)

Added `description` field to the node object to enable operators to store any information relates to the node. The field is limited to 4096 characters.

1.50 (Stein, 12.1.0)

Added `owner` field to the node object to enable operators to store information in relation to the owner of a node. The field is up to 255 characters and **MAY** be used in a later point in time to allow designation and deligation of permissions.

1.49 (Stein, 12.0.0)

Added new endpoints for retrieving conductors information, and added a `conductor` field to node object.

1.48 (Stein, 12.0.0)

Added `protected` field to the node object to allow protecting deployed nodes from undeploying, rebuilding or deletion. Also added `protected_reason` to specify the reason of making the node protected.

1.47 (Stein, 12.0.0)

Added `automated_clean` field to the node object, enabling cleaning per node.

1.46 (Rocky, 11.1.0)

Added `conductor_group` field to the node and the node response, as well as support to the API to return results by matching the parameter.

1.45 (Rocky, 11.1.0)

Added `reset_interfaces` parameter to nodes PATCH request, to specify whether to reset hardware interfaces to their defaults on drivers update.

1.44 (Rocky, 11.1.0)

Added `deploy_step` to the node object, to indicate the current deploy step (if any) being performed on the node.

1.43 (Rocky, 11.0.0)

Added `?detail=` boolean query to the API list endpoints to provide a more RESTful alternative to the existing `/nodes/detail` and similar endpoints.

1.42 (Rocky, 11.0.0)

Added `fault` to the node object, to indicate currently detected fault on the node.

1.41 (Rocky, 11.0.0)

Added support to abort inspection of a node in the `inspect wait` state.

1.40 (Rocky, 11.0.0)

Added BIOS properties as sub resources of nodes:

- GET `/v1/nodes/<node_ident>/bios`
- GET `/v1/nodes/<node_ident>/bios/<setting_name>`

Added `bios_interface` field to the node object to allow getting and setting the interface.

1.39 (Rocky, 11.0.0)

Added `inspect wait` to available provision states. A node is shown as `inspect wait` instead of `inspecting` during asynchronous inspection.

1.38 (Queens, 10.1.0)

Added `provision_state` verbs `rescue` and `unrescue` along with the following states: `rescue`, `rescue failed`, `rescue wait`, `rescuing`, `unrescue failed`, and `unrescuing`. After rescuing a node, it will be left in the `rescue` state running a rescue ramdisk, configured with the `rescue_password`, and listening with `ssh` on the specified network interfaces. Unrescuing a node will return it to `active`.

Added `rescue_interface` to the node object, to allow setting the rescue interface for a dynamic driver.

1.37 (Queens, 10.1.0)

Adds support for node traits, with the following new endpoints.

- GET `/v1/nodes/<node identifier>/traits` lists the traits for a node.
- PUT `/v1/nodes/<node identifier>/traits` sets all traits for a node.
- PUT `/v1/nodes/<node identifier>/traits/<trait>` adds a trait to a node.
- DELETE `/v1/nodes/<node identifier>/traits` removes all traits from a node.
- DELETE `/v1/nodes/<node identifier>/traits/<trait>` removes a trait from a node.

A nodes traits are also included the following node query and list responses:

- GET /v1/nodes/<node identifier>
- GET /v1/nodes/detail
- GET /v1/nodes?fields=traits

Traits cannot be specified on node creation, nor can they be updated via a PATCH request on the node.

1.36 (Queens, 10.0.0)

Added `agent_version` parameter to deploy heartbeat request for version negotiation with Ironic Python Agent features.

1.35 (Queens, 9.2.0)

Added ability to provide `configdrive` when node is updated to rebuild provision state.

1.34 (Pike, 9.0.0)

Adds a `physical_network` field to the port object. All ports in a portgroup must have the same value in their `physical_network` field.

1.33 (Pike, 9.0.0)

Added `storage_interface` field to the node object to allow getting and setting the interface.

Added `default_storage_interface` and `enabled_storage_interfaces` fields to the driver object to show the information.

1.32 (Pike, 9.0.0)

Added new endpoints for remote volume configuration:

- GET /v1/volume as a root for volume resources
- GET /v1/volume/connectors for listing volume connectors
- POST /v1/volume/connectors for creating a volume connector
- GET /v1/volume/connectors/<UUID> for showing a volume connector
- PATCH /v1/volume/connectors/<UUID> for updating a volume connector
- DELETE /v1/volume/connectors/<UUID> for deleting a volume connector
- GET /v1/volume/targets for listing volume targets
- POST /v1/volume/targets for creating a volume target
- GET /v1/volume/targets/<UUID> for showing a volume target
- PATCH /v1/volume/targets/<UUID> for updating a volume target

- DELETE /v1/volume/targets/<UUID> for deleting a volume target

Volume resources also can be listed as sub resources of nodes:

- GET /v1/nodes/<node identifier>/volume
- GET /v1/nodes/<node identifier>/volume/connectors
- GET /v1/nodes/<node identifier>/volume/targets

1.31 (Ocata, 7.0.0)

Added the following fields to the node object, to allow getting and setting interfaces for a dynamic driver:

- boot_interface
- console_interface
- deploy_interface
- inspect_interface
- management_interface
- power_interface
- raid_interface
- vendor_interface

1.30 (Ocata, 7.0.0)

Added dynamic driver APIs:

- GET /v1/drivers now accepts a `type` parameter (optional, one of `classic` or `dynamic`), to limit the result to only classic drivers or dynamic drivers (hardware types). Without this parameter, both classic and dynamic drivers are returned.
- GET /v1/drivers now accepts a `detail` parameter (optional, one of `True` or `False`), to show all fields for a driver. Defaults to `False`.
- GET /v1/drivers now returns an additional `type` field to show if the driver is classic or dynamic.
- GET /v1/drivers/<name> now returns an additional `type` field to show if the driver is classic or dynamic.
- GET /v1/drivers/<name> now returns additional fields that are null for classic drivers, and set as following for dynamic drivers:
 - The value of the `default_<interface-type>_interface` is the endpoint name of the calculated default interface for that type:
 - * `default_boot_interface`
 - * `default_console_interface`
 - * `default_deploy_interface`
 - * `default_inspect_interface`
 - * `default_management_interface`

- * `default_network_interface`
 - * `default_power_interface`
 - * `default_raid_interface`
 - * `default_vendor_interface`
- The value of the `enabled_<interface-type>_interfaces` is a list of endpoint names of the enabled interfaces for that type:
- * `enabled_boot_interfaces`
 - * `enabled_console_interfaces`
 - * `enabled_deploy_interfaces`
 - * `enabled_inspect_interfaces`
 - * `enabled_management_interfaces`
 - * `enabled_network_interfaces`
 - * `enabled_power_interfaces`
 - * `enabled_raid_interfaces`
 - * `enabled_vendor_interfaces`

1.29 (Ocata, 7.0.0)

Add a new management API to support inject NMI, PUT `/v1/nodes/(node_ident)/management/inject_nmi`.

1.28 (Ocata, 7.0.0)

Add `/v1/nodes/<node identifier>/vifs` endpoint for attach, detach and list of VIFs.

1.27 (Ocata, 7.0.0)

Add `soft rebooting` and `soft power off` as possible values for the `target` field of the power state change payload, and also add `timeout` field to it.

1.26 (Ocata, 7.0.0)

Add `portgroup mode` and `properties` fields.

1.25 (Ocata, 7.0.0)

Add possibility to unset chassis_uuid from a node.

1.24 (Ocata, 7.0.0)

Added new endpoints /v1/nodes/<node>/portgroups and /v1/portgroups/<portgroup>/ports. Added new field port.portgroup_uuid.

1.23 (Ocata, 7.0.0)

Added /v1/portgroups/ endpoint.

1.22 (Newton, 6.1.0)

Added endpoints for deployment ramdisks.

1.21 (Newton, 6.1.0)

Add node resource_class field.

1.20 (Newton, 6.1.0)

Add node network_interface field.

1.19 (Newton, 6.1.0)

Add local_link_connection and pxe_enabled fields to the port object.

1.18 (Newton, 6.1.0)

Add internal_info readonly field to the port object, that will be used by ironic to store internal port-related information.

1.17 (Newton, 6.0.0)

Addition of provision_state verb adopt which allows an operator to move a node from manageable state to active state without performing a deployment operation on the node. This is intended for nodes that have already been deployed by external means.

1.16 (Mitaka, 5.0.0)

Add ability to filter nodes by driver.

1.15 (Mitaka, 5.0.0)

Add ability to do manual cleaning when a node is in the manageable provision state via PUT `v1/nodes/<identifier>/states/provision, target:clean, clean_steps:[]`.

1.14 (Liberty, 4.2.0)

Make the following endpoints discoverable via Ironic API:

- `/v1/nodes/<UUID or logical name>/states`
- `/v1/drivers/<driver name>/properties`

1.13 (Liberty, 4.2.0)

Add a new verb `abort` to the API used to abort nodes in `CLEANWAIT` state.

1.12 (Liberty, 4.2.0)

This API version adds the following abilities:

- Get/set `node.target_raid_config` and to get `node.raid_config`.
- Retrieve the logical disk properties for the driver.

1.11 (Liberty, 4.0.0, breaking change)

Newly registered nodes begin in the `enroll` provision state by default, instead of `available`. To get them to the `available` state, the `manage` action must first be run to verify basic hardware control. On success the node moves to `manageable` provision state. Then the `provide` action must be run. Automated cleaning of the node is done and the node is made `available`.

1.10 (Liberty, 4.0.0)

Logical node names support all RFC 3986 unreserved characters. Previously only valid fully qualified domain names could be used.

1.9 (Liberty, 4.0.0)

Add ability to filter nodes by provision state.

1.8 (Liberty, 4.0.0)

Add ability to return a subset of resource fields.

1.7 (Liberty, 4.0.0)

Add node `clean_step` field.

1.6 (Kilo)

Add *Hardware Inspection* process: introduce `inspecting` and `inspectfail` provision states, and `inspect` action that can be used when a node is in `manageable` provision state.

1.5 (Kilo)

Add logical node names that can be used to address a node in addition to the node UUID. Name is expected to be a valid [fully qualified domain name](#) in this version of API.

1.4 (Kilo)

Add `manageable` state and `manage` transition, which can be used to move a node to `manageable` state from `available`. The node cannot be deployed in `manageable` state. This change is mostly a preparation for future inspection work and introduction of `enroll` provision state.

1.3 (Kilo)

Add node `driver_internal_info` field.

1.2 (Kilo, breaking change)

Renamed `NOSTATE` (`None` in Python, `null` in JSON) node state to `available`. This is needed to reduce confusion around `None` state, especially when future additions to the state machine land.

1.1 (Kilo)

This was the initial version when API versioning was introduced. Includes the following changes from Kilo release cycle:

- Add node `maintenance_reason` field and an API endpoint to set/unset the node maintenance mode.
- Add sync and async support for vendor passthru methods.
- Vendor passthru endpoints support different HTTP methods, not only POST.
- Make vendor methods discoverable via the Ironic API.
- Add logic to store the config drive passed by Nova.

This has been the minimum supported version since versioning was introduced.

1.0 (Juno)

This version denotes Juno API and was never explicitly supported, as API versioning was not implemented in Juno, and 1.1 became the minimum supported version in Kilo.

Exceptions from Versioning

The following API-visible things are not covered by the API versioning:

- Current node state is always exposed as it is, even if not supported by the requested API version, with exception of `available` state, which is returned in version 1.1 as `None` (in Python) or `null` (in JSON).
- Data within free-form JSON attributes: `properties`, `driver_info`, `instance_info`, `driver_internal_info` fields on a node object; extra fields on all objects.
- Addition of new drivers.
- All vendor passthru methods.

COMMAND REFERENCES

Here are references for commands not elsewhere documented.

7.1 Command References

Here are references for commands not elsewhere documented.

7.1.1 `ironic-dbsync`

The **`ironic-dbsync`** utility is used to create the database schema tables that the ironic services will use for storage. It can also be used to upgrade existing database tables when migrating between different versions of ironic.

The [Alembic library](#) is used to perform the database migrations.

Options

This is a partial list of the most useful options. To see the full list, run the following:

```
ironic-dbsync --help
```

-h, --help

Show help message and exit.

--config-dir <DIR>

Path to a config directory with configuration files.

--config-file <PATH>

Path to a configuration file to use.

-d, --debug

Print debugging output.

--version

Show the programs version number and exit.

**`upgrade`, `stamp`, `revision`, `version`, `create_schema`,
`online_data_migrations`**

The *command* to run.

Usage

Options for the various *commands* for **ironic-dbsync** are listed when the `-h` or `--help` option is used after the command.

For example:

```
ironic-dbsync create_schema --help
```

Information about the database is read from the ironic configuration file used by the API server and conductor services. This file must be specified with the `--config-file` option:

```
ironic-dbsync --config-file /path/to/ironic.conf create_schema
```

The configuration file defines the database backend to use with the *connection* database option:

```
[database]
connection=mysql+pymysql://root@localhost/ironic
```

If no configuration file is specified with the `--config-file` option, **ironic-dbsync** assumes an SQLite database.

Command Options

ironic-dbsync is given a command that tells the utility what actions to perform. These commands can take arguments. Several commands are available:

create_schema

-h, --help

Show help for `create_schema` and exit.

This command will create database tables based on the most current version. It assumes that there are no existing tables.

An example of creating database tables with the most recent version:

```
ironic-dbsync --config-file=/etc/ironic/ironic.conf create_schema
```

online_data_migrations

-h, --help

Show help for `online_data_migrations` and exit.

--max-count <NUMBER>

The maximum number of objects (a positive value) to migrate. Optional. If not specified, all the objects will be migrated (in batches of 50 to avoid locking the database for long periods of time).

--option <MIGRATION.KEY=VALUE>

If a migration accepts additional parameters, they can be passed via this argument. It can be specified several times.

This command will migrate objects in the database to their most recent versions. This command must be successfully run (return code 0) before upgrading to a future release.

It returns:

- 1 (not completed) if there are still pending objects to be migrated. Before upgrading to a newer release, this command must be run until 0 is returned.
- 0 (success) after migrations are finished or there are no data to migrate
- 127 (error) if max-count is not a positive value or an option is invalid
- 2 (error) if the database is not compatible with this release. This command needs to be run using the previous release of ironic, before upgrading and running it with this release.

revision

-h, --help

Show help for revision and exit.

-m <MESSAGE>, --message <MESSAGE>

The message to use with the revision file.

--autogenerate

Compares table metadata in the application with the status of the database and generates migrations based on this comparison.

This command will create a new revision file. You can use the `--message` option to comment the revision.

This is really only useful for ironic developers making changes that require database changes. This revision file is used during database migration and will specify the changes that need to be made to the database tables. Further discussion is beyond the scope of this document.

stamp

-h, --help

Show help for stamp and exit.

--revision <REVISION>

The revision number.

This command will stamp the revision table with the version specified with the `--revision` option. It will not run any migrations.

upgrade

-h, --help

Show help for upgrade and exit.

--revision <REVISION>

The revision number to upgrade to.

This command will upgrade existing database tables to the most recent version, or to the version specified with the `--revision` option.

Before this upgrade is invoked, the command **ironic-dbsync online_data_migrations** must have been successfully run using the previous version of ironic (if you are doing an upgrade as opposed to a new installation of ironic). If it wasn't run, the database will not be compatible with this recent version of ironic, and this command will return 2 (error).

If there are no existing tables, then new tables are created, beginning with the oldest known version, and successively upgraded using all of the database migration files, until they are at the specified version. Note that this behavior is different from the `create_schema` command that creates the tables based on the most recent version.

An example of upgrading to the most recent table versions:

```
ironic-dbsync --config-file=/etc/ironic/ironic.conf upgrade
```

Note: This command is the default if no command is given to **ironic-dbsync**.

Warning: The upgrade command is not compatible with SQLite databases since it uses ALTER TABLE commands to upgrade the database tables. SQLite supports only a limited subset of ALTER TABLE.

version

-h, --help

Show help for version and exit.

This command will output the current database version.

7.1.2 ironic-status

Synopsis

```
ironic-status <category> <command> [<args>]
```


Description

ironic-status is a tool that provides routines for checking the status of a Ironic deployment.

Options

The standard pattern for executing a **ironic-status** command is:

```
ironic-status <category> <command> [<args>]
```

Run without arguments to see a list of available command categories:

```
ironic-status
```

Categories are:

- upgrade

Detailed descriptions are below.

You can also run with a category argument such as `upgrade` to see a list of all commands in that category:

```
ironic-status upgrade
```

These sections describe the available categories and arguments for **ironic-status**.

Upgrade

ironic-status upgrade check Performs a release-specific readiness check before restarting services with new code. This command expects to have complete configuration and access to databases and services.

Return Codes

Return code	Description
0	All upgrade readiness checks passed successfully and there is nothing to do.
1	At least one check encountered an issue and requires further investigation. This is considered a warning but the upgrade may be OK.
2	There was an upgrade status check failure that needs to be investigated. This should be considered something that stops an upgrade.
255	An unexpected error occurred.

History of Checks

12.0.0 (Stein)

- Adds a check for compatibility of the object versions with the release of ironic.

Wallaby

- Adds a check to validate the configured policy file is not JSON based as JSON based policies have been deprecated.

CONTRIBUTOR GUIDE

8.1 Developers Guide

8.1.1 Getting Started

If you are new to ironic, this section contains information that should help you get started as a developer working on the project or contributing to the project.

So You Want to Contribute

This document provides some necessary points for developers to consider when writing and reviewing Ironic code. The checklist will help developers get things right.

Getting Started

If you're completely new to OpenStack and want to contribute to the ironic project, please start by familiarizing yourself with the [Infra Teams Developer Guide](#). This will help you get your accounts set up in Launchpad and Gerrit, familiarize you with the workflow for the OpenStack continuous integration and testing systems, and help you with your first commit.

LaunchPad

Most of the tools used for OpenStack require a launchpad.net ID for authentication. Ironic previously used to track work on Launchpad, but we have not done so since migrating to Storyboard.

See also:

- <https://launchpad.net>

Storyboard

The ironic project moved from Launchpad to [Storyboard](#) for work and task tracking. This provides an aggregate view called a Project Group and individual Projects. A good starting place is the [project group](#) representing the whole of the ironic community, as opposed to the [ironic project](#) storyboard which represents ironic as a repository.

See *Bug Reporting and Triaging Guide* for more details on how we track bugs.

Internet Relay Chat IRC

Daily contributor discussions take place on IRC in the #openstack-ironic channel on the OFTC IRC network.

Please feel free to join us at <ircs://irc.oftc.net:6697> and join our channel!

Additional information on getting connected can be found in the [OpenStack community contribution guide](#).

Everything Ironic

Ironic is a community of projects centered around the primary project repository ironic, which help facilitate the deployment and management of bare metal resources.

This means there are a number of different repositories that fall into the responsibility of the project team and the community. Some of the repositories may not seem strictly hardware related, but they may be tools or things to just make an aspect easier.

Related Projects

There are several projects that are tightly integrated with ironic and which are developed by the same community.

See also:

- [Bifrost Documentation](#)
- [Ironic Inspector Documentation](#)
- [Ironic Lib Documentation](#)
- [Ironic Python Agent \(IPA\) Documentation](#)
- [Ironic Client Documentation](#)
- [Ironic Inspector Client Documentation](#)

Useful Links

Bug/Task tracker <https://storyboard.openstack.org/#!/project/943>

Mailing list (prefix Subject line with [ironic]) <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-discuss>

Code Hosting <https://opendev.org/openstack/ironic>

Code Review <https://review.opendev.org/#/q/status:open+project:openstack/ironic,n,z>

Whiteboard <https://etherpad.openstack.org/p/IronicWhiteBoard>

Weekly Meeting Agenda https://wiki.openstack.org/wiki/Meetings/Ironic#Agenda_for_next_meeting

Adding New Features

Ironic tracks new features using RFEs (Requests for Feature Enhancements) instead of blueprints. These are stories with rfe tag, and they should be submitted before a spec or code is proposed.

When a member of the [ironic-core team](#) decides that the proposal is worth implementing, a spec (if needed) and code should be submitted, referencing the RFE task or story ID number. Contributors are welcome to submit a spec and/or code before the RFE is approved, however those patches will not land until the RFE is approved.

Feature Submission Process

1. Submit a bug report on the [ironic StoryBoard](#). There are two fields that must be filled: Title and Description. Tasks can be added and are associated with a project. If you cant describe it in a sentence or two, it may mean that you are either trying to capture more than one RFE at once, or that you are having a hard time defining what you are trying to solve at all. This may also be a sign that your feature may require a specification document.
2. Describe the proposed change in the Description field. The description should provide enough details for a knowledgeable developer to understand what is the existing problem in the current platform that needs to be addressed, or what is the enhancement that would make the platform more capable, both from a functional and a non-functional standpoint.
3. Submit the story, add an rfe tag to it and assign yourself or whoever is going to work on this feature.
4. As soon as a member of the team acknowledges the story, we will move the story to the Review state. As time goes on, Discussion about the RFE, and whether to approve it will occur.
5. Contributors will evaluate the RFE and may advise the submitter to file a spec in the ironic-specs repository to elaborate on the feature request. Typically this is when an RFE requires extra scrutiny, more design discussion, etc. For the spec submission process, please see the [Ironic Specs Process](#). A specific task should be created to track the creation of a specification.
6. If a spec is not required, once the discussion has happened and there is positive consensus among the ironic-core team on the RFE, the RFE is approved, and its tag will move from rfe to rfe-approved. This means that the feature is approved and the related code may be merged.
7. If a spec is required, the spec must be submitted (with a new task as part of the story referenced as Task in the commit message), reviewed, and merged before the RFE will be approved (and the tag changed to rfe-approved).

8. The tasks then goes through the usual process first to Review when the spec/code is being worked on, then Merged when it is implemented.
9. If the RFE is rejected, the ironic-core team will move the story to Invalid status.

Change Tracking

We track our stories and tasks in Storyboard.

<https://storybook.openstack.org/#!/project/ironic>

When working on an RFE, please be sure to tag your commits properly: Story: #xxxx or Task: #xxxx. It is also helpful to set a consistent review topic, such as story/xxxx for all patches related to the RFE.

If the RFE spans across several projects (e.g. ironic and python-ironicclient), but the main work is going to happen within ironic, please use the same story for all the code youre submitting, there is no need to create a separate RFE in every project.

Note: RFEs may only be approved by members of the ironic-core team.

Note: While not strictly required for minor changes and fixes, it is highly preferred by the Ironic community that any change which needs to be backported, have a recorded Story and Task in Storyboard.

Managing Change Sets

If you would like some help, or if you (or some members of your team) are unable to continue working on the feature, updating and maintaining the changes, please let the rest of the ironic community know. You could leave a comment in one or more of the changes/patches, bring it up in IRC, the weekly meeting, or on the OpenStack development email list. Communicating this will make other contributors aware of the situation and allow for others to step forward and volunteer to continue with the work.

In the event that a contributor leaves the community, do not expect the contributors changes to be continued unless someone volunteers to do so.

Getting Your Patch Merged

Within the Ironic project, we generally require two core reviewers to sign-off (+2) change sets. We also will generally recognize non-core (+1) reviewers, and sometimes even reverse our decision to merge code based upon their reviews.

We recognize that some repositories have less visibility, as such it is okay to ask for a review in our IRC channel. Please be prepared to stay in IRC for a little while in case we have questions.

Sometimes we may also approve patches with a single core reviewer. This is generally discouraged, but sometimes necessary. When we do so, we try to explain why we do so. As a patch submitter, it equally helps us to understand why the change is important. Generally, more detail and context helps us understand the change faster.

Timeline Expectations

As with any large project, it does take time for features and changes to be merged in any of the project repositories. This is largely due to limited review bandwidth coupled with varying reviewer priorities and focuses.

When establishing an understanding of complexity, the following things should be kept in mind.

- Generally, small and minor changes can gain consensus and merge fairly quickly. These sorts of changes would be: bug fixes, minor documentation updates, follow-up changes.
- Medium changes generally consist of driver feature parity changes, where one driver is working to match functionality of another driver.
 - These changes generally only require an RFE for the purposes of tracking and correlating the change.
 - Documentation updates are expected to be submitted with or immediately following the initial change set.
- Larger or controversial changes generally take much longer to merge. This is often due to the necessity of reviewers to gain additional context and for change sets to be iterated upon to reach a state where there is consensus. These sorts of changes include: database, object, internal interface additions, RPC, rest API changes.
 - These changes will very often require specifications to reach consensus, unless there are pre-existing patterns or code already present.
 - These changes may require many reviews and iterations, and can also expect to be impacted by merge conflicts as other code or features are merged.
 - These changes must typically be split into a series of changes. Reviewers typically shy away from larger single change sets due to increased difficulty in reviewing.
 - Do not expect any API or user-visible data model changes to merge after the API client freeze. Some substrate changes may merge if not user visible.
- You should expect complex features, such as cross-project features or integration, to take longer than a single development cycle to land.
 - Building consensus is vital.
 - Often these changes are controversial or have multiple considerations that need to be worked through in the specification process, which may cause the design to change. As such, it may take months to reach consensus over design.
 - These features are best broken into larger chunks and tackled in an incremental fashion.

Live Upgrade Related Concerns

See *Rolling Upgrades*.

Driver Internal Info

The `driver_internal_info` node field was introduced in the Kilo release. It allows driver developers to store internal information that can not be modified by end users. Here is the list of existing common and agent driver attributes:

- Common attributes:
 - `is_whole_disk_image`: A Boolean value to indicate whether the user image contains ramdisk/kernel.
 - `clean_steps`: An ordered list of clean steps that will be performed on the node.
 - `deploy_steps`: An ordered list of deploy steps that will be performed on the node. Support for deploy steps was added in the 11.1.0 release.
 - `instance`: A list of dictionaries containing the disk layout values.
 - `root_uuid_or_disk_id`: A String value of the bare metal nodes root partition uuid or disk id.
 - `persistent_boot_device`: A String value of device from `ironic.common.boot_devices`.
 - `is_next_boot_persistent`: A Boolean value to indicate whether the next boot device is `persistent_boot_device`.
- Agent driver attributes:
 - `agent_url`: A String value of IPA API URL so that Ironic can talk to IPA ramdisk.
 - `hardware_manager_version`: A String value of the version of the hardware manager in IPA ramdisk.
 - `target_raid_config`: A Dictionary containing the target RAID configuration. This is a copy of the same name attribute in Node object. But this one is never actually saved into DB and is only read by IPA ramdisk.

Note: These are only some fields in use. Other vendor drivers might expose more `driver_internal_info` properties, please check their development documentation and/or module docstring for details. It is important for developers to make sure these properties follow the precedent of prefixing their variable names with a specific interface name (e.g., `ilo_bar`, `drac_xyz`), so as to minimize or avoid any conflicts between interfaces.

Ironic Specs Process

Specifications must follow the template which can be found at [specs/template.rst](#), which is quite self-documenting. Specifications are proposed by adding them to the *specs/approved* directory, adding a soft link to it from the *specs/not-implemented* directory, and posting it for review to Gerrit. For more information, please see the [README](#).

The same [Gerrit process](#) as with source code, using the repository [ironic-specs](#), is used to add new specifications.

All approved specifications are available at: <https://specs.openstack.org/openstack/ironic-specs>. If a specification has been approved but not completed within one or more releases since the approval, it may be re-reviewed to make sure it still makes sense as written.

Ironic specifications are part of the *RFE (Requests for Feature Enhancements) process*. You are welcome to submit patches associated with an RFE, but they will have a -2 (do not merge) until the specification has been approved. This is to ensure that the patches don't get accidentally merged beforehand. You will still be able to get reviewer feedback and push new patch sets, even with a -2. The [list of core reviewers](#) for the specifications is small but mighty. (This is not necessarily the same list of core reviewers for code patches.)

Changes to existing specs

For approved but not-completed specs:

- cosmetic cleanup, fixing errors, and changing the definition of a feature can be done to the spec.

For approved and completed specs:

- changing a previously approved and completed spec should only be done for cosmetic cleanup or fixing errors.
- changing the definition of the feature should be done in a new spec.

Please see the [Ironic specs process wiki page](#) for further reference.

Bug Reporting

Bugs can be reported via our Task and Bug tracking tool Storyboard.

When filing bugs, please include as much detail as possible, and don't be shy.

Essential pieces of information are generally:

- Contents of the node - *baremetal node show <uuid>*
- Steps to reproduce the issue.
- Exceptions and surrounding lines from the logs.
- Versions of ironic, ironic-python-agent, and any other coupled components.

Please also set your expectations of what *should* be happening. Statements of user expectations are how we understand what is occurring and how we learn new use cases!

Project Team Leader Duties

The Project Team Leader or PTL is elected each development cycle by the contributors to the ironic community.

Think of this person as your primary contact if you need to try and rally the project, or have a major issue that requires attention.

They serve a role that is mainly oriented towards trying to drive the technical discussion forward and managing the idiosyncrasies of the project. With this responsibility, they are considered a public face of the project and are generally obliged to try and provide project updates and outreach communication.

All common PTL duties are enumerated here in the [PTL guide](#).

Tasks like release management or preparation for a release are generally delegated with-in the team. Even outreach can be delegated, and specifically there is no rule stating that any member of the community cant propose a release, clean-up release notes or documentation, or even get on the occasional stage.

Bug Reporting and Triaging Guide

Storyboard

All ironic projects use [Storyboard](#) for tracking both bugs and enhancement requests (RFE). The [ironic project group](#) lists all our projects.

Note: Ironic is developed as part of OpenStack and therefore uses the `openstack/` namespace.

Storyboard is somewhat different from traditional bug tracking systems because every *story* is not linked to a project itself, but rather through its *tasks*. A story represents an issue you are facing or an enhancement you want to see, while tasks represent individual action items which can span several projects. When creating a story, youll also need to create the first task. If unsure, create a task against `openstack/ironic`.

Reporting Guide

We are constantly receiving a lot of requests, so its important to file a meaningful story for it to be acted upon. A good story:

- specifies **why** a change is needed.
- explains how to reproduce the described condition.

Note: Please try to provide a reproducer based on unit tests, [devstack](#) or [bifrost](#). While we try our best to support users using other installers and distributions, it may be non-trivial without deep knowledge of them. If youre using a commercial distribution or a product, please try contacting support first.

- should be understandable without additional context. For example, if you see an exception, we will need the full traceback.

- should not be too verbose either. Unfortunately, we cannot process a few days worth of system logs to find the problems, we expect your collaboration.
- is not a question or a support request. Please see *So You Want to Contribute* for the ways to contact us.
- provides a way to contact the reporter. Please follow the comments and expect follow-up emails, but ideally also be on IRC for questions.

An enhancement request additionally:

- benefits the overall project, not just one consumer. If you have a case that is specific to your requirements, think about ways to make ironic extensible to be able to cover it.
- does not unnecessary increase the project scope. Consider if your idea can be implemented without changing ironic or its projects, maybe it actually should?

Triaging Guide

The bug triaging process involves checking new stories to make sure they are actionable by the team. This guide is mostly targeting the project team, but we would appreciate if reporters could partly self-triage their own requests.

- Determine if the request is valid and complete. Use the checklist in the *Reporting Guide* for that.
- Is the request a bug report or an enhancement request (an RFE)? The difference is often subtle, the key question to answer is if the described behavior is expected.

Add an `rfe` tag to all enhancement requests and propose it for the RFE Review section of the [weekly meeting](#).

- Does the RFE obviously require a `spec`? Usually this is decided when an RFE is reviewed during the meeting, but some requests are undoubtedly complex, involve changing a lot of critical parts and thus demand a spec.

Add a `needs-spec` tag to enhancement requests that obviously need a spec. Otherwise leave it until the meeting.

- Apply additional tags:
 - All hardware type specific stories should receive a corresponding tag (e.g. `ipmi`, `idrac`, etc).
 - API-related stories should have an `api` tag.
 - CI issues should have a `gate` tag.

The next actions **must only** be done by a core team member (or an experienced full-time contributor appointed by the PTL):

- Can the RFE be automatically approved? It happens if the RFE requests an implementation of a driver feature that is already implemented for other drivers and does not pose additional complexity.
If the RFE can be automatically approved, apply the `rfe-approved` tag. If unsure, never apply the tag! Talk to the PTL instead.
- Does the RFE have a corresponding spec approved? If yes, apply the `rfe-approved` tag.
- In the end, apply the `ironic-triaged` tag to make the story as triaged.

Expiring Bugs

While we hope to fix all issues that our consumers hit, it is unfortunately not realistic. Stories **may** be closed by marking all their tasks `INVALID` in the following cases:

- No solution has been proposed in 1 calendar year.
- Additional information has been requested from the reporter, and no update has been provided in 1 calendar month.
- The request no longer aligns with the direction of the project.

Note: As usual, common sense should be applied when closing stories.

Developer Quick-Start

This is a quick walkthrough to get you started developing code for Ironic. This assumes you are already familiar with submitting code reviews to an OpenStack project.

The gate currently runs the unit tests under Python 3.6, 3.7, and 3.8. It is strongly encouraged to run the unit tests locally prior to submitting a patch.

Note: Do not run unit tests on the same environment as devstack due to conflicting configuration with system dependencies.

Note: This document is compatible with Python (3.6, 3.7, 3.8), Debian buster (10.8), Ubuntu Focal Fossa (20.04 LTS), RHEL8/CentOS Stream, openSUSE/SLE 15, and Fedora (33). When referring to different versions of Python and OS distributions, this is explicitly stated.

See also:

<https://docs.openstack.org/infra/manual/developers.html#development-workflow>

Prepare Development System

System Prerequisites

The following packages cover the prerequisites for a local development environment on most current distributions. Instructions for getting set up with non-default versions of Python and on older distributions are included below as well.

- Ubuntu/Debian:

```
sudo apt-get install build-essential python3-dev libssl-dev python3-pip  
↳ libmysqlclient-dev libxml2-dev libxslt-dev libpq-dev git git-review  
↳ libffi-dev gettext ipmitool psmisc graphviz libjpeg-dev
```

- RHEL/CentOS/Fedora:

```
sudo dnf install python3-devel openssl-devel python3-pip mysql-devel_
↳ libxml2-devel libxslt-devel postgresql-devel git git-review libffi-
↳ devel gettext ipmitool psmisc graphviz gcc libjpeg-turbo-devel
```

- openSUSE/SLE:

```
sudo zypper install git git-review libffi-devel libmysqlclient-devel_
↳ libopenssl-devel libxml2-devel libxslt-devel postgresql-devel python3-
↳ devel python-nose python3-pip gettext-runtime psmisc
```

To run the tests locally, it is a requirement that your terminal emulator supports unicode with the `en_US.UTF8` locale. If you use `locale-gen` to manage your locales, make sure you have enabled `en_US.UTF8` in `/etc/locale.gen` and rerun `locale-gen`.

Python Prerequisites

We suggest to use at least `tox 3.9`, if your distribution has an older version, you can install it using `pip` system-wise or better per user using the `user` option that by default will install the binary under `$HOME/.local/bin`, so you need to be sure to have that path in `$PATH`; for example:

```
pip install tox --user
```

will install `tox` as `~/.local/bin/tox`

You may need to explicitly upgrade `virtualenv` if youve installed the one from your OS distribution and it is too old (`tox` will complain). You can upgrade it individually, if you need to:

```
pip install -U virtualenv --user
```

Running Unit Tests Locally

If you havent already, Ironic source code should be pulled directly from git:

```
# from your home or source directory
cd ~
git clone https://opendev.org/openstack/ironic
cd ironic
```

Running Unit and Style Tests

All unit tests should be run using `tox`. To run Ironics entire test suite:

```
# to run the py3 unit tests, and the style tests
tox
```

To run a specific test or tests, use the `-e` option followed by the `tox` target name. For example:

```
# run the unit tests under py36 and also run the pep8 tests
tox -epy36 -epep8
```

You may pass options to the test programs using positional arguments. To run a specific unit test, this passes the desired test (regex string) to `stestr`:

```
# run a specific test for Python 3.6
tox -epy36 -- test_conductor
```

Debugging unit tests

In order to break into the debugger from a unit test we need to insert a breaking point to the code:

```
import pdb; pdb.set_trace()
```

Then run `tox` with the debug environment as one of the following:

```
tox -e debug
tox -e debug test_file_name
tox -e debug test_file_name.TestClass
tox -e debug test_file_name.TestClass.test_name
```

For more information see the [oslotest documentation](#).

Database Setup

The unit tests need a local database setup, you can use `tools/test-setup.sh` to set up the database the same way as setup in the OpenStack test systems.

Additional Tox Targets

There are several additional tox targets not included in the default list, such as the target which builds the documentation site. See the `tox.ini` file for a complete listing of tox targets. These can be run directly by specifying the target name:

```
# generate the documentation pages locally
tox -edocs

# generate the sample configuration file
tox -egenconfig
```

Exercising the Services Locally

In addition to running automated tests, sometimes it can be helpful to actually run the services locally, without needing a server in a remote datacenter.

If you would like to exercise the Ironic services in isolation within your local environment, you can do this without starting any other OpenStack services. For example, this is useful for rapidly prototyping and debugging interactions over the RPC channel, testing database migrations, and so forth.

Here we describe two ways to install and configure the dependencies, either run directly on your local machine or encapsulated in a virtual machine or container.

Step 1: Create a Python virtualenv

1. If you havent already downloaded the source code, do that first:

```
cd ~
git clone https://opendev.org/openstack/ironic
cd ironic
```

2. Create the Python virtualenv:

```
tox -eenv --notest --develop -r
```

3. Activate the virtual environment:

```
. .tox/venv/bin/activate
```

4. Install the *openstack* client command utility:

```
pip install python-openstackclient
```

5. Install the *baremetal* client:

```
pip install python-ironicclient
```

Note: You can install `python-ironicclient` from source by cloning the git repository and running `pip install .` while in the root of the cloned repository.

6. Export some ENV vars so the client will connect to the local services that youll start in the next section:

```
export OS_AUTH_TYPE=none
export OS_ENDPOINT=http://localhost:6385/
```

Next, install and configure system dependencies.

Step 2: Install System Dependencies Locally

This step will install MySQL on your local system. This may not be desirable in some situations (eg, youre developing from a laptop and do not want to run a MySQL server on it all the time). If you want to use SQLite, skip it and do not set the `connection` option.

1. Install `mysql-server`:

Ubuntu/Debian:

```
sudo apt-get install mysql-server
```

RHEL/CentOS/Fedora:

```
sudo dnf install mariadb mariadb-server
sudo systemctl start mariadb.service
```

openSUSE/SLE:: sudo zypper install mariadb sudo systemctl start mysql.service

If using MySQL, you need to create the initial database:

```
mysql -u root -pMYSQL_ROOT_PWD -e "create schema ironic"
```

Note: if you choose not to install mysql-server, ironic will default to using a local sqlite database. The database will then be stored in `ironic/ironic.sqlite`.

2. Create a configuration file within the ironic source directory:

```
# generate a sample config
tox -egenconfig

# copy sample config and modify it as necessary
cp etc/ironic/ironic.conf.sample etc/ironic/ironic.conf.local

# disable auth since we are not running keystone here
sed -i "s/#auth_strategy = keystone/auth_strategy = noauth/" etc/ironic/
↪ironic.conf.local

# use the 'fake-hardware' test hardware type
sed -i "s/#enabled_hardware_types = */enabled_hardware_types = fake-
↪hardware/" etc/ironic/ironic.conf.local

# use the 'fake' deploy and boot interfaces
sed -i "s/#enabled_deploy_interfaces = */enabled_deploy_interfaces = ↪
↪fake/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_boot_interfaces = */enabled_boot_interfaces = fake/" ↪
↪etc/ironic/ironic.conf.local

# enable both fake and ipmitool management and power interfaces
sed -i "s/#enabled_management_interfaces = */enabled_management_
↪interfaces = fake,ipmitool/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_power_interfaces = */enabled_power_interfaces = fake,
↪ipmitool/" etc/ironic/ironic.conf.local

# change the periodic sync_power_state_interval to a week, to avoid ↪
↪getting NodeLocked exceptions
sed -i "s/#sync_power_state_interval = 60/sync_power_state_interval = ↪
↪604800/" etc/ironic/ironic.conf.local

# if you opted to install mysql-server, switch the DB connection from ↪
↪sqlite to mysql
sed -i "s/#connection = */connection = mysql\+pymysql:\|\/\|root:MYSQL_
↪ROOT_PWD@localhost\/ironic/" etc/ironic/ironic.conf.local

# use JSON RPC to avoid installing rabbitmq locally
sed -i "s/#rpc_transport = oslo/rpc_transport = json-rpc/" etc/ironic/
↪ironic.conf.local
```


Step 3: Start the Services

From within the python virtualenv, run the following command to prepare the database before you start the ironic services:

```
# initialize the database for ironic
ironic-dbsync --config-file etc/ironic/ironic.conf.local create_schema
```

Next, open two new terminals for this section, and run each of the examples here in a separate terminal. In this way, the services will *not* be run as daemons; you can observe their output and stop them with Ctrl-C at any time.

1. Start the API service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-api -d --config-file etc/ironic/ironic.conf.local
```

2. Start the Conductor service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-conductor -d --config-file etc/ironic/ironic.conf.local
```

Step 4: Interact with the running services

You should now be able to interact with ironic via the python client, which is present in the python virtualenv, and observe both services debug outputs in the other two windows. This is a good way to test new features or play with the functionality without necessarily starting DevStack.

To get started, export the following variables to point the client at the local instance of ironic and disable the authentication:

```
export OS_AUTH_TYPE=none
export OS_ENDPOINT=http://127.0.0.1:6385
```

Then list the available commands and resources:

```
# get a list of available commands
openstack help baremetal

# get the list of drivers currently supported by the available conductor(s)
baremetal driver list

# get a list of nodes (should be empty at this point)
baremetal node list
```

Here is an example walkthrough of creating a node:

```
MAC="aa:bb:cc:dd:ee:ff" # replace with the MAC of a data port on your node
IPMI_ADDR="1.2.3.4"     # replace with a real IP of the node BMC
```

(continues on next page)

(continued from previous page)

```
IPMI_USER="admin"          # replace with the BMC's user name
IPMI_PASS="pass"          # replace with the BMC's password

# enroll the node with the fake hardware type and IPMI-based power and
# management interfaces. Note that driver info may be added at node
# creation time with "--driver-info"
NODE=$(baremetal node create \
    --driver fake-hardware \
    --management-interface ipmitool \
    --power-interface ipmitool \
    --driver-info ipmi_address=$IPMI_ADDR \
    --driver-info ipmi_username=$IPMI_USER \
    -f value -c uuid)

# driver info may also be added or updated later on
baremetal node set $NODE --driver-info ipmi_password=$IPMI_PASS

# add a network port
baremetal port create $MAC --node $NODE

# view the information for the node
baremetal node show $NODE

# request that the node's driver validate the supplied information
baremetal node validate $NODE

# you have now enrolled a node sufficiently to be able to control
# its power state from ironic!
baremetal node power on $NODE
```

If you make some code changes and want to test their effects, simply stop the services with Ctrl-C and restart them.

Step 5: Fixing your test environment

If you are testing changes that add or remove python entrypoints, or making significant changes to ironics python modules, or simply keep the virtualenv around for a long time, your development environment may reach an inconsistent state. It may help to delete cached .pyc files, update dependencies, reinstall ironic, or even recreate the virtualenv. The following commands may help with that, but are not an exhaustive troubleshooting guide:

```
# clear cached pyc files
cd ~/ironic/ironic
find ./ -name '*.pyc' | xargs rm

# reinstall ironic modules
cd ~/ironic
. .tox/venv/bin/activate
pip uninstall ironic
```

(continues on next page)

(continued from previous page)

```
pip install -e .

# install and upgrade ironic and all python dependencies
cd ~/ironic
. .tox/venv/bin/activate
pip install -U -e .
```

Deploying Ironic with DevStack

DevStack may be configured to deploy Ironic, setup Nova to use the Ironic driver and provide hardware resources (network, baremetal compute nodes) using a combination of OpenVSwitch and libvirt. It is highly recommended to deploy on an expendable virtual machine and not on your personal work station. Deploying Ironic with DevStack requires a machine running Ubuntu 16.04 (or later) or Fedora 24 (or later). Make sure your machine is fully up to date and has the latest packages installed before beginning this process.

The `ironic-tempest-plugin` is necessary if you want to run integration tests, the section *Ironic with ironic-tempest-plugin* tells the extra steps you need to enable it in DevStack.

See also:

<https://docs.openstack.org/devstack/latest/>

Note: The devstack demo tenant is now granted the `baremetal_observer` role and thereby has read-only access to ironics API. This is sufficient for all the examples below. Should you want to create or modify bare metal resources directly (ie. through ironic rather than through nova) you will need to use the devstack admin tenant.

Devstack will no longer create the user stack with the desired permissions, but does provide a script to perform the task:

```
git clone https://opendev.org/openstack/devstack.git devstack
sudo ./devstack/tools/create-stack-user.sh
```

Note: In case you receive an error `Could not determine host ip address`. See `local.conf` for suggestions on setting `HOST_IP`, you need to manually add the main ip of your machine to the `localrc` file under `devstack/` using the `HOST_IP` variable, e.g. `HOST_IP=YOURIP` This could happen when running devstack on virtual machines.

Switch to the stack user and clone DevStack:

```
sudo su - stack
git clone https://opendev.org/openstack/devstack.git devstack
```

Ironic

Create `devstack/local.conf` with minimal settings required to enable Ironic. An example `local.conf` that enables the `direct` *deploy interface* and uses the `ipmi` hardware type by default:

```
cd devstack
cat >local.conf <<END
[[local|localrc]]
# Enable only minimal services
disable_all_services
enable_service g-api
enable_service key
enable_service memory_tracker
enable_service mysql
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service q-metering
enable_service q-svc
enable_service rabbit

# Credentials
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic

# Create 3 virtual machines to pose as Ironic's baremetal nodes.
IRONIC_VM_COUNT=3
IRONIC_BAREMETAL_BASIC_OPS=True
DEFAULT_INSTANCE_TYPE=baremetal

IRONIC_RPC_TRANSPORT=json-rpc
IRONIC_RAMDISK_TYPE=tinyipa

# Enable additional hardware types, if needed.
#IRONIC_ENABLED_HARDWARE_TYPES=ipmi,fake-hardware
# Don't forget that many hardware types require enabling of additional
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
```

(continues on next page)

(continued from previous page)

```

IRONIC_DEPLOY_DRIVER="ipmi"

# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=1024
IRONIC_VM_SPECS_DISK=3

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0

# To build your own IPA ramdisk from source, set this to True
IRONIC_BUILD_DEPLOY_RAMDISK=False

INSTALL_TEMPEST=False
VIRT_DRIVER=ironic

# By default, DevStack creates a 10.0.0.0/24 network for instances.
# If this overlaps with the hosts network, you may adjust with the
# following.
IP_VERSION=4
FIXED_RANGE=10.1.0.0/20
IPV4_ADDRS_SAFE_TO_USE=10.1.0.0/20
NETWORK_GATEWAY=10.1.0.1

Q_AGENT=openvswitch
Q_ML2_PLUGIN_MECHANISM_DRIVERS=openvswitch
Q_ML2_TENANT_NETWORK_TYPE=vxlan

# Log all output to files
LOGFILE=/opt/stack/devstack.log
LOGDIR=/opt/stack/logs
IRONIC_VM_LOG_DIR=/opt/stack/ironic-bm-logs

END

```

Ironic with ironic-tempest-plugin

Using the stack user, clone the ironic-tempest-plugin repository in the same directory you cloned DevStack:

```
git clone https://opendev.org/openstack/ironic-tempest-plugin.git
```

An example local.conf that enables the ironic tempest plugin and Ironic can be found below. The TEMPEST_PLUGINS variable needs to have the absolute path to the ironic-tempest-plugin folder, otherwise the plugin won't be installed. Ironic will have enabled the *direct deploy interface* and uses the ipmi hardware type by default:

```
cd devstack
```

(continues on next page)

(continued from previous page)

```
cat >local.conf <<END
[[local|localrc]]
# Credentials
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
SWIFT_HASH=password
SWIFT_TEMPURL_KEY=password

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic

# Disable nova novnc service, ironic does not support it anyway.
disable_service n-novnc

# Enable Swift for the direct deploy interface.
enable_service s-proxy
enable_service s-object
enable_service s-container
enable_service s-account

# Disable Horizon
disable_service horizon

# Disable Cinder
disable_service cinder c-sch c-api c-vol

# Swift temp URL's are required for the direct deploy interface
SWIFT_ENABLE_TEMPURLS=True

# Create 3 virtual machines to pose as Ironic's baremetal nodes.
IRONIC_VM_COUNT=3
IRONIC_BAREMETAL_BASIC_OPS=True
DEFAULT_INSTANCE_TYPE=baremetal

# Enable additional hardware types, if needed.
#IRONIC_ENABLED_HARDWARE_TYPES=ipmi,fake-hardware
# Don't forget that many hardware types require enabling of additional
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
IRONIC_DEPLOY_DRIVER=ipmi
```

(continues on next page)

(continued from previous page)

```
# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=2048
IRONIC_VM_SPECS_DISK=10

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0

# To build your own IPA ramdisk from source, set this to True
IRONIC_BUILD_DEPLOY_RAMDISK=False

VIRT_DRIVER=ironic

# By default, DevStack creates a 10.0.0.0/24 network for instances.
# If this overlaps with the hosts network, you may adjust with the
# following.
NETWORK_GATEWAY=10.1.0.1
FIXED_RANGE=10.1.0.0/24
FIXED_NETWORK_SIZE=256

# Log all output to files
LOGFILE=$HOME/devstack.log
LOGDIR=$HOME/logs
IRONIC_VM_LOG_DIR=$HOME/ironic-bm-logs
TEMPEST_PLUGINS="/opt/stack/ironic-tempest-plugin"

END
```

Note: Some tests may be skipped depending on the configuration of your environment, they may be reliant on a driver or a capability that you did not configure.

Deployment

Note: Git protocol requires access to port 9418, which is not a standard port that corporate firewalls always allow. If you are behind a firewall or on a proxy that blocks Git protocol, modify the `enable_plugin` line to use `https://` instead of `git://` and add `GIT_BASE=https://opendev.org` to the credentials:

```
GIT_BASE=https://opendev.org

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic
```

Note: When the `ipmi` hardware type is used and `IRONIC_IS_HARDWARE` variable is `false` devstack

will automatically set up [VirtualBMC](#) to control the power state of the virtual baremetal nodes.

Note: When running QEMU as non-root user (e.g. `qemu` on Fedora or `libvirt-qemu` on Ubuntu), make sure `IRONIC_VM_LOG_DIR` points to a directory where QEMU will be able to write. You can verify this with, for example:

```
# on Fedora
sudo -u qemu touch $HOME/ironic-bm-logs/test.log
# on Ubuntu
sudo -u libvirt-qemu touch $HOME/ironic-bm-logs/test.log
```

Note: To check out an in-progress patch for testing, you can add a Git ref to the `enable_plugin` line. For instance:

```
enable_plugin ironic https://opendev.org/openstack/ironic refs/changes/46/
↪295946/15
```

For a patch in review, you can find the ref to use by clicking the Download button in Gerrit. You can also specify a different git repo, or a branch or tag:

```
enable_plugin ironic https://github.com/openstack/ironic stable/kilo
```

For more details, see the [devstack plugin interface documentation](#).

Run `stack.sh`:

```
./stack.sh
```

Source credentials, create a key, and spawn an instance as the demo user:

```
. ~/devstack/openrc

# query the image id of the default cirros image
image=$(openstack image show $DEFAULT_IMAGE_NAME -f value -c id)

# create keypair
ssh-keygen
openstack keypair create --public-key ~/.ssh/id_rsa.pub default

# spawn instance
openstack server create --flavor baremetal --image $image --key-name default_
↪testing
```

Note: Because devstack create multiple networks, we need to pass an additional parameter `--nic net-id` to the nova boot command when using the admin account, for example:


```
net_id=$(openstack network list | egrep "$PRIVATE_NETWORK_NAME"'^-]' | awk '
↳{ print $2 }')

openstack server create --flavor baremetal --nic net-id=$net_id --image
↳$image --key-name default testing
```

You should now see a Nova instance building:

```
openstack server list --long
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| ID          | Name      | Status | Task State | Power State | Networks | Image_
↳Name | Image ID | Availability Zone | Host | Properties |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| a2c7f812 | testing | BUILD  | spawning   | NOSTATE     |          | cirros-
↳0.3 | 44d4092a | nova   |            |             |          | .5-x86_
| -e386-4a |         |        |            |             |          | disk
↳64- | -51ac-47 |         |            |             |          |
| 22-b393- |         |        |            |             |          |
↳   | 51-9c50- |         |            |             |          |
| fe1802ab |         |        |            |             |          |
↳   | fd6e2050 |         |            |             |          |
| d56e     |         |        |            |             |          |
↳   | faa1     |         |            |             |          |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
```

Nova will be interfacing with Ironic conductor to spawn the node. On the Ironic side, you should see an Ironic node associated with this Nova instance. It should be powered on and in a wait call-back provisioning state:

```
baremetal node list
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| UUID          | Name      | Instance UUID |
↳   | Power State | Provisioning State | Maintenance |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| 9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f | node-0 | None
↳   | power off   | None      | False      |
| ec0c6384-cc3a-4edf-b7db-abde1998be96 | node-1 | None
↳   | power off   | None      | False      |
| 4099e31c-576c-48f8-b460-75e1b14e497f | node-2 | a2c7f812-e386-4a22-b393-
↳fe1802abd56e | power on    | wait call-back | False      |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
```

At this point, Ironic conductor has called to libvirt (via virtualbmc) to power on a virtual machine, which will PXE + TFTP boot from the conductor node and progress through the Ironic provisioning workflow.

One libvirt domain should be active now:

```
sudo virsh list --all
Id      Name                               State
-----
 2      node-2                             running
-       node-0                             shut off
-       node-1                             shut off
```

This provisioning process may take some time depending on the performance of the host system, but Ironic should eventually show the node as having an active provisioning state:

```
baremetal node list
+-----+-----+-----+-----+
| UUID                                     | Name   | Instance UUID |
|-----+-----+-----+-----+
| 9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f | node-0 | None          |
|-----+-----+-----+-----+
| ec0c6384-cc3a-4edf-b7db-abde1998be96 | node-1 | None          |
|-----+-----+-----+-----+
| 4099e31c-576c-48f8-b460-75e1b14e497f | node-2 | a2c7f812-e386-4a22-b393-
| fe1802abd56e | power on   | active        | False          |
+-----+-----+-----+-----+
```

This should also be reflected in the Nova instance state, which at this point should be ACTIVE, Running and an associated private IP:

```
openstack server list --long
+-----+-----+-----+-----+-----+-----+
| ID          | Name   | Status | Task State | Power State | Networks |
|-----+-----+-----+-----+-----+-----+
| a2c7f812 | testing | ACTIVE | none       | Running     | private=10.1. |
|-----+-----+-----+-----+-----+-----+
| cirros-0.3 | 44d4092a | nova  |           |             |          |
|-----+-----+-----+-----+-----+-----+
| -e386-4a | -51ac-47 |       |           |             | 0.4, fd7d:1f3 | .5-
|-----+-----+-----+-----+-----+-----+
| x86_64- | 22-b393- |       |           |             | c:4bf1:0:f816 |
|-----+-----+-----+-----+-----+-----+
| disk      | 51-9c50- |       |           |             | :3eff:f39d:6d |
|-----+-----+-----+-----+-----+-----+
| fe1802ab | fd6e2050 |       |           |             |          |
|-----+-----+-----+-----+-----+-----+
| d56e     | faa1    |       |           |             | 94          |
|-----+-----+-----+-----+-----+-----+
```

The server should now be accessible via SSH:

```
ssh cirros@10.1.0.4
$
```

Running Tempest tests

After *Deploying Ironic with DevStack* with the `ironic-tempest-plugin` enabled, one might want to run integration tests against the running cloud. The Tempest project is the project that offers an integration test suite for OpenStack.

First, navigate to Tempest directory:

```
cd /opt/stack/tempest
```

To run all tests from the [Ironic plugin](#), execute the following command:

```
tox -e all -- ironic
```

To limit the amount of tests that you would like to run, you can use a regex. For instance, to limit the run to a single test file, the following command can be used:

```
tox -e all -- ironic_tempest_plugin.tests.scenario.test_baremetal_basic_ops
```

Debugging Tempest tests

It is sometimes useful to step through the test code, line by line, especially when the error output is vague. This can be done by running the tests in debug mode and using a debugger such as [pdb](#).

For example, after editing the `test_baremetal_basic_ops` file and setting up the `pdb` traces you can invoke the `run_tempest.sh` script in the Tempest directory with the following parameters:

```
./run_tempest.sh -N -d ironic_tempest_plugin.tests.scenario.test_baremetal_
↪basic_ops
```

- The `-N` parameter tells the script to run the tests in the local environment (without a `virtualenv`) so it can find the Ironic tempest plugin.
- The `-d` parameter enables the debug mode, allowing it to be used with `pdb`.

For more information about the supported parameters see:

```
./run_tempest.sh --help
```

Note: Always be careful when running debuggers in time sensitive code, they may cause timeout errors that weren't there before.

OSProfiler Tracing in Ironic

OSProfiler is an OpenStack cross-project profiling library. It is being used among OpenStack projects to look at performance issues and detect bottlenecks. For details on how OSProfiler works and how to use it in ironic, please refer to [OSProfiler Support Documentation](#).

Building developer documentation

If you would like to build the documentation locally, eg. to test your documentation changes before uploading them for review, run these commands to build the documentation set:

- On your local machine:

```
# activate your development virtualenv
. .tox/venv/bin/activate

# build the docs
tox -edocs

#Now use your browser to open the top-level index.html located at:

ironic/doc/build/html/index.html
```

- On a remote machine:

```
# Go to the directory that contains the docs
cd ~/ironic/doc/source/

# Build the docs
tox -edocs

# Change directory to the newly built HTML files
cd ~/ironic/doc/build/html/

# Create a server using python on port 8000
python -m SimpleHTTPServer 8000

#Now use your browser to open the top-level index.html located at:

http://your_ip:8000
```

Developer FAQ (frequently asked questions)

Here are some answers to frequently-asked questions from IRC and elsewhere.

- *How do I*
 - *create a migration script template?*
 - *know if a release note is needed for my change?*
 - *create a new release note?*
 - *update a release note?*
 - *get a decision on something?*
 - *add support for GMRs to new executables and extending the GMR?*

How do I

create a migration script template?

Using the `ironic-dbsync revision` command, e.g:

```
$ cd ironic
$ tox -e env -- ironic-dbsync revision -m "create foo table"
```

It will create an empty alembic migration. For more information see the [alembic documentation](#).

know if a release note is needed for my change?

[Reno documentation](#) contains a description of what can be added to each section of a release note. If, after reading this, you're still unsure about whether to add a release note for your change or not, keep in mind that it is intended to contain information for deployers, so changes to unit tests or documentation are unlikely to require one.

create a new release note?

By running `reno` command via `tox`, e.g:

```
$ tox -e venv -- reno new version-foo
venv create: /home/foo/ironic/.tox/venv
venv installdeps: -r/home/foo/ironic/test-requirements.txt
venv develop-inst: /home/foo/ironic
venv runtests: PYTHONHASHSEED='0'
venv runtests: commands[0] | reno new version-foo
Created new notes file in releasenotes/notes/version-foo-ecb3875dc1cbf6d9.
↪yaml
venv: commands succeeded
congratulations :)
```

(continues on next page)

(continued from previous page)

```
$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)

releasenotes/notes/version-foo-ecb3875dc1cbf6d9.yaml
```

Then edit the result file. Note that:

- we prefer to use present tense in release notes. For example, a release note should say Adds support for feature foo, not Added support for feature foo. (We use adds instead of add because grammatically, it is ironic adds support, not ironic add support.)
- any variant of English spelling (American, British, Canadian, Australian) is acceptable. The release note itself should be consistent and not have different spelling variants of the same word.

For more information see the [reno documentation](#).

update a release note?

If this is a release note that pertains to something that was fixed on master or an intermediary release (during a development cycle, that hasnt been branched yet), you can go ahead and update it by submitting a patch.

If it is the release note of an ironic release that has branched, [it can be updated](#) but we will only allow it in extenuating circumstances. (It can be updated by *only* updating the file in that branch. DO NOT update the file in master and cherry-pick it. If you do, [see how the mess was cleaned up](#).)

get a decision on something?

You have an issue and would like a decision to be made. First, make sure that the issue hasnt already been addressed, by looking at documentation, stories, specifications, or asking. Information and links can be found on the [Ironic wiki](#) page.

There are several ways to solicit comments and opinions:

- bringing it up at the [weekly Ironic meeting](#)
- bringing it up on [IRC](#)
- bringing it up on the [mailing list](#) (add [Ironic] to the Subject of the email)

If there are enough core folks at the weekly meeting, after discussing an issue, voting could happen and a decision could be made. The problem with IRC or the weekly meeting is that feedback will only come from the people that are actually present.

To inform (and solicit feedback from) more people about an issue, the preferred process is:

1. bring it up on the mailing list
2. after some period of time has elapsed (and depending on the thread activity), someone should propose a solution via gerrit. (E.g. the person that started the thread if no one else steps up.) The

proposal should be made in the git repository that is associated with the issue. (For instance, this decision process was proposed as a documentation patch to the ironic repository.)

3. In the email thread, dont forget to provide a link to the proposed patch!
4. The discussion then moves to the proposed patch. If this is a big decision, we could declare that some percentage of the cores should vote on it before landing it.

(This process was suggested in an email thread about [process for making decisions](#).)

add support for GMRs to new executables and extending the GMR?

For more information, see the [oslo.reports documentation](#) page.

Contributor Vision

Background

During the Rocky Project Teams Gathering (February/March 2018), The contributors in the room at that time took a few minutes to write out each contributors vision of where they see ironic in five years time.

After everyone had a chance to spend a few minutes writing, we went around the room and gave every contributor the chance to read their vision and allow other contributors to ask questions to better understand what each individual contributor wrote. While we were doing that, we also took time to capture the common themes.

This entire exercise did result in some laughs and a common set of words, and truly helped to ensure that the entire team proceeded to use the same words to describe various aspects as the sessions progressed during the week. We also agreed that we should write a shared vision, to have something to reference and remind us of where we want to go as a community.

Rocky Vision: For 2022-2023

Common Themes

Below is an entirely unscientific summary of common themes that arose during the discussion among fourteen contributors.

- Contributors picked a time between 2020, and 2023.
- 4 Contributors foresee ironic being the leading Open Source baremetal deployment technology
- 2 Contributors foresee ironic reaching feature parity with Nova.
- 2 Contributors foresee users moving all workloads to the cloud
- 1 Contributor foresees Kubernetes and Container integration being the major focus of Bare Metal as a Service further down the road.
- 2 Contributors foresee greater composable hardware being more common.
- 1 Contributor foresees ironic growing into or supporting CMDBs.
- 2 Contributors foresee that features are more micro-service oriented.

- 2 Contributors foresee that ironic supported all of the possible baremetal management needs
- 1 Contributor foresees standalone use being more common.
- 2 Contributors foresee the ironics developer community growing
- 2 Contributors foresee that auto-discovery will be more common.
- 2 Contributors foresee ironic being used for devices beyond servers, such as lightbulbs, IOT, etc.

Vision Statement

The year is 2022. Were meeting to plan the Z release of Ironic. We stopped to reflect upon the last few years of Ironics growth, how we had come such a long way to become the defacto open source baremetal deployment technology. How we had grown our use cases, and support for consumers such as containers, and users who wished to managed specialized fleets of composed machines.

New contributors and their different use cases have brought us closer to parity with virtual machines. Everyday were gaining word of more operators adopting the ironic communitys CMDB integration to leverage hardware discovery. Weve heard of operators deploying racks upon racks of new hardware by just connecting the power and network cables, and from there the operators have discovered time to write the worlds greatest operator novel with the time saved in commissioning new racks of hardware.

Time has brought us closer and taught us to be more collaborative across the community, and we look forward to our next release together.

Comparison to the 2018 OpenStack Technical Vision

In late-2018, the OpenStack Technical composed a [technical vision](#) of what OpenStack clouds should look like. While every component differs, and cloudy interactions change dramatically the closer to physical hardware one gets, there are a few areas where Ironic could use some improvement.

This list is largely for the purposes of help wanted. It is also important to note that Ironic as a project has a [vision document](#) for itself.

The Pillars of Cloud - Self Service

- Ironics mechanisms and tooling are low level infrastructure mechanisms and as such there has never been a huge emphasis or need on making Ironic be capable of offering direct multi-tenant interaction. Most users interact with the bare metal managed by Ironic via Nova, which abstracts away many of these issues. Eventually, we should offer direct multi-tenancy which is not oriented towards admin-only.

Design Goals - Built-in Reliability and Durability

- Ironic presently considers in-flight operations as failed upon the restart of a controller that was previously performing a task, because we do not know the current status of the task upon re-start. In some cases, this makes sense, but potentially requires administrative intervention in the worst of cases. In a perfect universe, Ironic conductors would validate their perception, in case tasks actually finished.

Design Goals - Graphical User Interface

- While a graphical interface was developed for Horizon in the form of `ironic-ui`, currently `ironic-ui` receives only minimal housekeeping. As Ironic has evolved, `ironic-ui` is stuck on version `1.34` and knows nothing of our evolution since. Ironic ultimately needs a contributor with sufficient time to pick up `ironic-ui` or to completely replace it as a functional and customizable user interface.

The following pages describe the architecture of the Bare Metal service and may be helpful to anyone working on or with the service, but are written primarily for developers.

System Architecture

High Level description

An Ironic deployment will be composed of the following components:

- An admin-only RESTful [API service](#), by which privileged users, such as cloud operators and other services within the cloud control plane, may interact with the managed bare metal servers.
- A [Conductor service](#), which does the bulk of the work. Functionality is exposed via the [API service](#). The Conductor and API services communicate via RPC.
- A Database and [DB API](#) for storing the state of the Conductor and Drivers.
- A Deployment Ramdisk or Deployment Agent, which provide control over the hardware which is not available remotely to the Conductor. A ramdisk should be built which contains one of these agents, eg. with [diskimage-builder](#). This ramdisk can be booted on-demand.

Note: The agent is never run inside a tenant instance.

Drivers

The internal driver API provides a consistent interface between the Conductor service and the driver implementations. A driver is defined by a *hardware type* deriving from the `AbstractHardwareType` class, defining supported *hardware interfaces*. See [Enabling drivers and hardware types](#) for a more detailed explanation. See [Pluggable Drivers](#) for an explanation on how to write new hardware types and interfaces.

Driver-Specific Periodic Tasks

Drivers may run their own periodic tasks, i.e. actions run repeatedly after a certain amount of time. Such a task is created by using the `periodic` decorator on an interface method. For example

```
from futurist import periodics

class FakePower(base.PowerInterface):
    @periodics.periodic(spacing=42)
    def task(self, manager, context):
        pass # do something
```

Here the `spacing` argument is a period in seconds for a given periodic task. For example `spacing=5` means every 5 seconds.

Driver-Specific Steps

Drivers may have specific steps that may need to be executed or offered to a user to execute in order to perform specific configuration tasks.

These steps should ideally be located on the management interface to enable consistent user experience of the hardware type. What should be avoided is duplication of existing interfaces such as the deploy interface to enable vendor specific cleaning or deployment steps.

Message Routing

Each Conductor registers itself in the database upon start-up, and periodically updates the timestamp of its record. Contained within this registration is a list of the drivers which this Conductor instance supports. This allows all services to maintain a consistent view of which Conductors and which drivers are available at all times.

Based on their respective driver, all nodes are mapped across the set of available Conductors using a [consistent hashing algorithm](#). Node-specific tasks are dispatched from the API tier to the appropriate conductor using conductor-specific RPC channels. As Conductor instances join or leave the cluster, nodes may be remapped to different Conductors, thus triggering various driver actions such as take-over or clean-up.

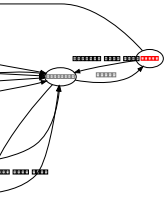
Ironics State Machine

State Machine Diagram

The diagram below shows the provisioning states that an Ironic node goes through during the lifetime of a node. The diagram also depicts the events that transition the node to different states.

Stable states are highlighted with a thicker border. All transitions from stable states are initiated by API requests. There are a few other API-initiated-transitions that are possible from non-stable states. The events for these API-initiated transitions are indicated with (via API). Internally, the conductor initiates the other transitions (depicted in gray).

State Descriptions



ated using API version 1.11 or newer. When a node is in the `enroll` state, the only thing ironic knows about it is that it exists, and ironic cannot take any further action by itself. Once a node has its driver/interfaces and their required information set in `node.driver_info`, the node can be transitioned to the `verifying` state by setting the nodes provision state using the `manage` verb.

using the information given in `node.driver_info` and with either the driver/hardware type and interfaces it has been assigned. This involves going out and confirming that the credentials work to access whatever node control mechanism they talk to.

node using the driver/interfaces and credentials passed in at node create time, the node will be transitioned to the `manageable` state. From `manageable`, nodes can transition to:

enroll (sta

This
is
the
state
that
all
node
start
off
in
when
cre-

verifying

iron
will
val-
i-
date
that
it
can
man
age
the
node

manageab

Once
iron
has
ver-
i-
fied
that
it
can
man
age
the

using the `clean` verb.

using the `inspect` verb.

by setting the nodes provision state using the `provide` verb.

- man...
(thro...
cle...
by...
set-
ting...
the...
node...
pro-
vi-
sion...
state
- man...
(thro...
insp...
by...
set-
ting...
the...
node...
pro-
vi-
sion...
state
- ava...
(thro...
cle...
if...
au-
to-
mati...
clea...
ing...
is...
en-
able
- act...
(thro...
adop...
by...
set-
ting...
the

using the adopt verb.

updates need to be made to it such as changes to fields in driver_info and updates to networking information on ironic ports assigned to the node.

tioned to, from these failure states:

node
pro-
vi-
sion
state

man.
is
the
state
that
a
node
shou
be
mov
into
whe
any

man.
is
also
the
only
sta-
ble
state
that
can
be
tran-
si-

- adop
fai
- cle
fai
- ins
fai

inspecting
ins
will
uti-

hardware-derived node properties to reflect the current state of the hardware. Typically, the node will transition to `manageable` if inspection is synchronous, or `inspect wait` if asynchronous. The node will transition to `inspect failed` if error occurred.

inspection is in progress. A successfully inspected node shall transition to `manageable` state.

tion of the node fails. From here the node can transitioned to:

lize
node
in-
tro-
spec
tion
to
up-
date

inspect wa

This
is
the
pro-
vi-
sion
state
used
when
an
asyn
chro

inspect fai

This
is
the
state
a
node
will
mov
into
when
in-
spec

- insp
by
set-
ting
the
node
pro-
vi-
sion

the inspect verb.

the manage verb

grammed into a known configuration.

ductor is executing the clean step (for out-of-band clean steps) or preparing the environment (building PXE configuration files, configuring the DHCP, etc) to boot the ramdisk for running in-band clean steps.

state
us-
ing

- man
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing

cleaning

Nod
in
the
cle
state
are
be-
ing
scru
and
re-
pro-

Whe
a
node
is
in
the
cle
state
it
mea
that
the
con-

clean wait

Just

are being scrubbed and reprogrammed. The difference is that in the `clean wait` state the conductor is waiting for the ramdisk to boot or the `clean` step which is running in-band to finish.

can be interrupted by setting the nodes provision state using the `abort` verb if the task that is running allows it.

and cleaned, they are moved into the `available` state and are ready to be provisioned. From `available`, nodes can transition to:

like
the
clea
state
the
node
in
the
clea
wai
state

The
clea
ing
pro-
cess
of
a
node
in
the
clea
wai
state

available

Af-
ter
node
have
been
suc-
cess
fully
pre-
con-
fig-
ured

- act.
(thro
dep
by
set-
ting

using the active verb.

the manage verb

load on them. This consists of running a series of tasks, such as:

the
node
pro-
vi-
sion
state

- man-
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing

deploying

Nod
in
dep
are
be-
ing
pre-
pare
to
run
a
worl

- Set-
ting
ap-
pro-
pri-
ate
BIO
con-
fig-
u-
ra-
tions

work config, a config drive partition, etc.) that may be required by additional subsystems.

ing deployed. The difference is that in `wait call-back` the conductor is waiting for the ramdisk to boot or execute parts of the deployment which need to run in-band on the node (for example, installing the bootloader, or writing the image to the disk).

Parti-
tion-
ing
drive
and
lay-
ing
dow
file
sys-
tems

- Cre-
at-
ing
any
ad-
di-
tion
re-
sour
(nod
spec
net-

wait call-b

Just
like
the
dep
state
the
node
in
wai
cal
are
be-

The
de-
ploy
men
of
a
node
in

errupted by setting the nodes provision state using the `deleted` verb.

ployment fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

using either the `active` or `rebuild` verbs.

sion state using the `deleted` verb.

wai
cal
can
be
in-

deploy fail

This
is
the
state
a
node
will
mov
into
whe
a
de-

- act.
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state

- ava.
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-

may collect out-of-band sensor information (including power state) on a regular basis. Nodes in active can transition to:

sion state using the deleted verb.

using the rebuild verb.

active (sta
Nod
in
act.
have
a
worl
load
run-
ning
on
them
iron

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-

- act.
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state

- res
(thro
res
by
set-
ting

using the `rescue` verb.

an active workload. In `deleting`, ironic tears down and removes any configuration and resources it added in `deploying` or `rescuing`.

an active deployment fails. From `error`, nodes can transition to:

sion state using the `deleted` verb.

the
node
pro-
vi-
sion
state

deleting

Node
in
del.
state
are
be-
ing
torn
down
from
run-
ning

error (stal

This
is
the
state
a
node
will
mov
into
when
dele
ing

- ava.
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-

a baremetal node with an existing workload on it. Ordinarily when a baremetal node is enrolled and managed by ironic, it must transition through `cleaning` and `deploying` to reach `active` state. However, those baremetal nodes that have an existing workload on them, do not need to be deployed or cleaned again, so this transition allows these nodes to move directly from `manageable` to `active`.

cue operations. This consists of running a series of tasks, such as:

adopting

This state allows ironic to take over management of

rescuing

Nodes in `rescuing` are being prepared to perform rescue

- Setting appropriate BIOS configurations
- Creating any additional

work config, etc.) that may be required by additional subsystems.

ing rescued. The difference is that in `rescue wait` the conductor is waiting for the ramdisk to boot or execute parts of the rescue which need to run in-band on the node (for example, setting the password for user named `rescue`).

can be aborted by setting the nodes provision state using the `abort` verb.

di-
tiona
re-
sour
(nod
spec
net-

rescue wait

Just
like
the
res
state
the
node
in
res
wai
are
be-

The
res-
cue
op-
er-
a-
tion
of
a
node
in
res
wai

rescue fail

This
is
the
state
a
node
will
mov
into
whe
a

cue operation fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

using the `rescue` verb.

using the `unrescue` verb.

using the `deleted` verb.

res-

- `rescue` (through the node provision state)

- `act` (through the node provision state)

- `ava` (through the node provision state)

rescue (state)
Node in rescue

Ironic may collect out-of-band sensor information (including power state) on a regular basis. Nodes in rescue can transition to:

using the `unrescue` verb.

using the `deleted` verb.

have
a
res-
cue
rame
run-
ning
on
them

- act.
(thro
unr
by
set-
ting
the
node
pro-
vi-
sion
state

- ava
(thro
del
by
set-
ting
the
node
pro-
vi-
sion
state

unrescuin

Nod
in
unr
are
be-
ing
pre-
pare
to
tran-

to active state from rescue state. This consists of running a series of tasks, such as setting appropriate BIOS configurations such as changing boot device.

rescue operation fails. From here the node can be transitioned to:

using the rescue verb.

using the unrescue verb.

si-
tion

unrescue

This
is
the
state
a
node
will
mov
into
whe
an
un-

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state

- act.
(thro
unr
by
set-
ting
the
node
pro-
vi-
sion
state

- ava.
(thro

using the `deleted` verb.

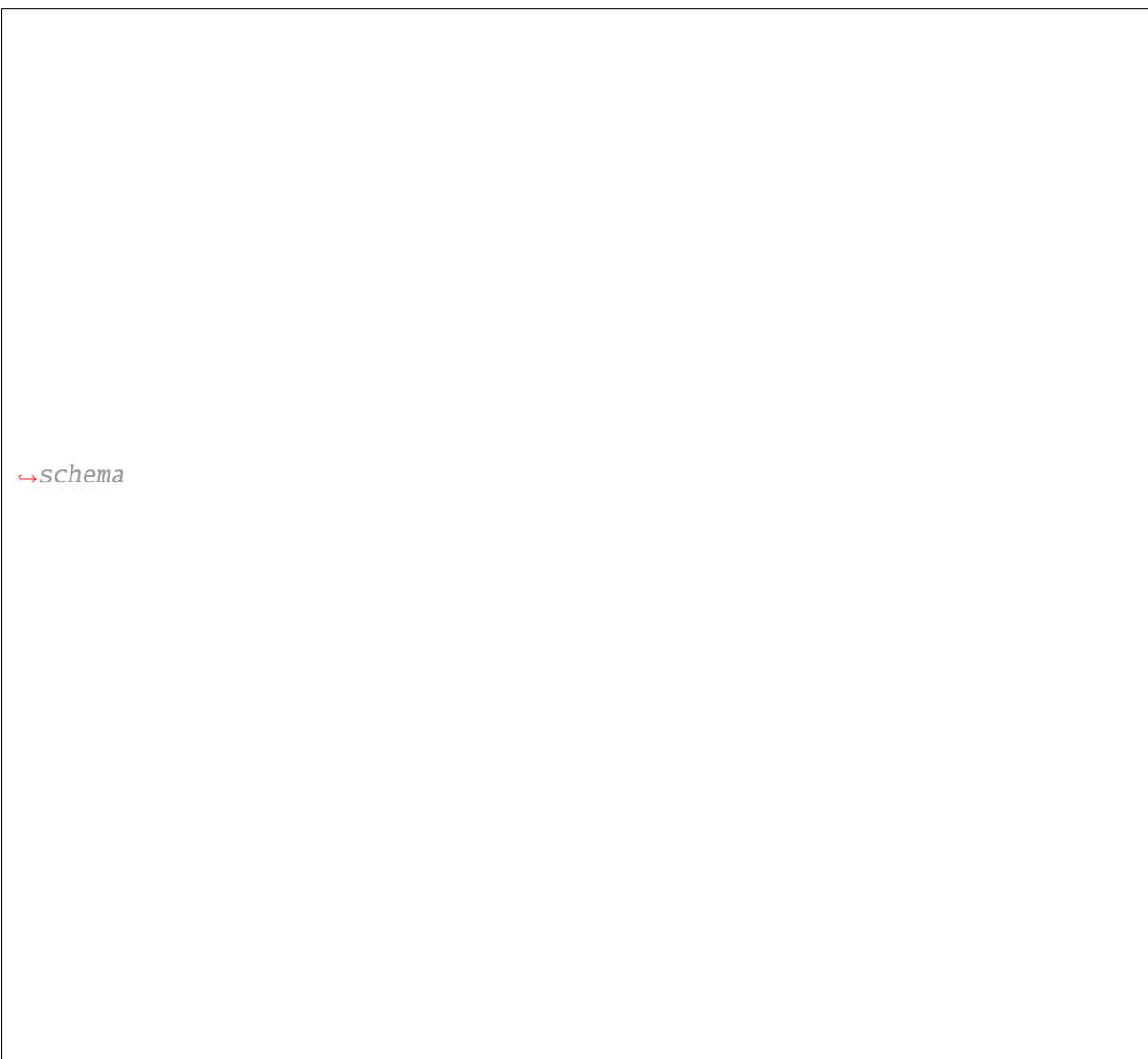
Developing New Notifications

tion by external services. Notifications are sent to these services over a message bus by `oslo.messaging` `Notifier` class. For more information about configuring notifications and available notifications, see *Notifications*.

defining the notification itself, the payload, and the other fields not auto-generated by oslo (`level`, `event_type` and `publisher_id`). Below describes how to use these base classes to add a new notification to ironic.

Adding a new notification to ironic

versioned notification class should be created by subclassing the NotificationBase class to define the notification itself and the NotificationPayloadBase class to define which fields the new notification will contain inside its payload. You may also define a schema to allow the payload to be automatically populated by the fields of an ironic object. Heres an example:



↪ *schema*

(continues on next page)

To
add
a
new
no-
ti-
fi-
ca-
tion
to
iron
a
new

↪ #
↪ T
↪ i
↪ o
↪ w
↪ f
↪ y
↪ w
↪ t
↪ u
↪ i
↪ y

@ba
↪ I
↪ r
cla
↪ E
↪ I

↪
↪
↪
↪
↪
↪ #
↪ V
↪ 1
↪ 0
↪ I
↪ V

(continued from previous page)

```
↪fields.StringField()
```

(continues on next page)

(continued from previous page)

```
→ 'ExampleNotifPayload')
```

(continues on next page)

(continued from previous page)

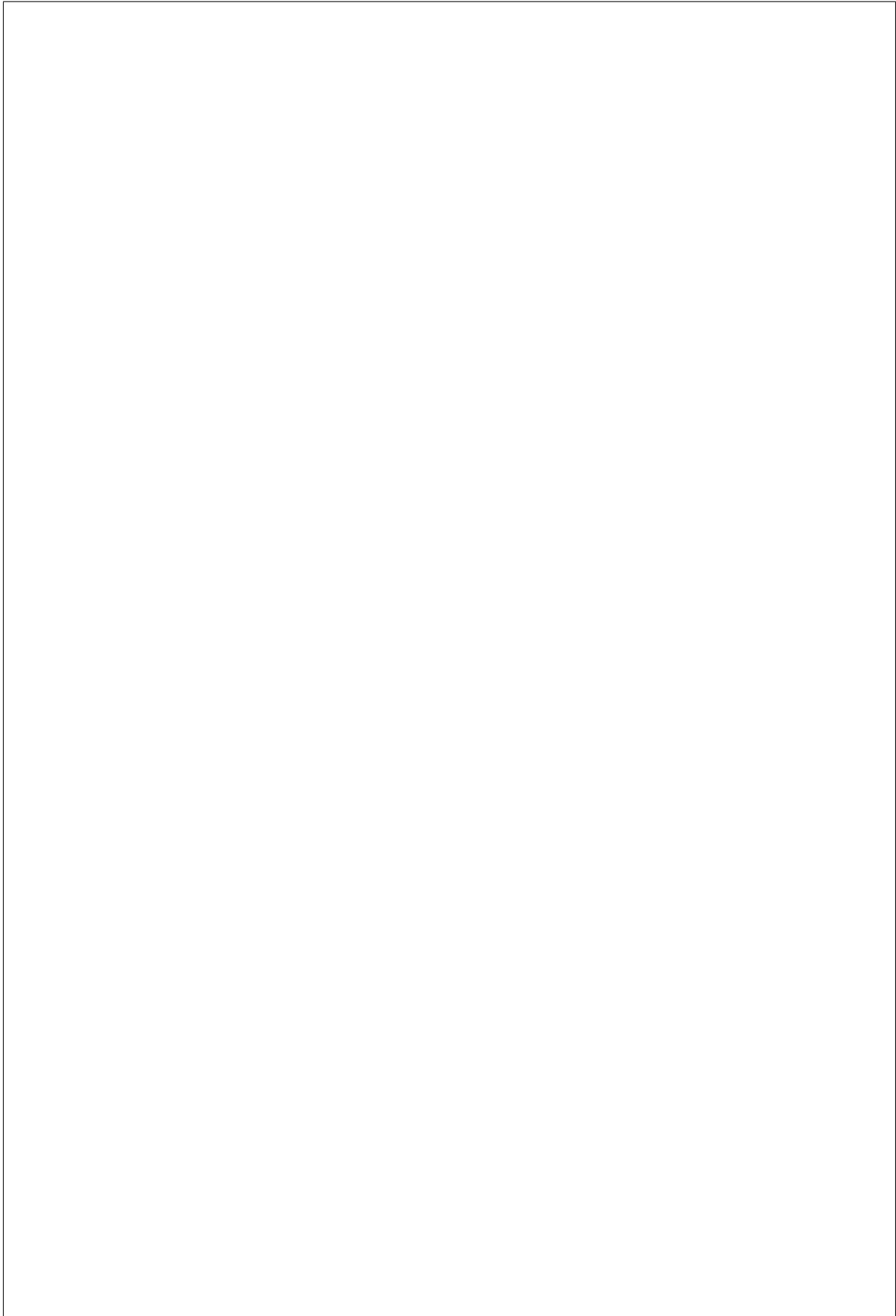
↳ calling `populate_schema` with

↳

↳ `('example_obj', 'a_useful_field')`

(continues on next page)

(continued from previous page)



`↪fields.StringField()`,

(continues on next page)

payload object; this field has to be defined as a field of the payload. The `<data_source_name>` shall refer to name of the parameter passed as kwarg to the payloads `populate_schema()` call and this object will be used as the source of the data. The `<field_of_the_data_source>` shall be a valid field of the passed argument.

the notification can be emitted.

`<field_of_the_data_source>` field. The `<data_source_name>` will not be part of the payload object internal or external representation.

in the same way as in any versioned object.

store
in
the

The
SCH
need
to
be
ap-
plied
with
the
pop
call
be-
fore

The
valu
of
the
pay:
<pa
field
will
be
set
by
the
<da

Pay-
load
field
that
are
not
set
by
the
SCH
can
be
filled

the following. Note that if you choose to define a schema in the SCHEMA class variable, you must populate the schema by calling `populate_schema(example_obj=my_example_obj)` before emitting the notification is allowed:

```
→ a_useful_field='important',  
  
→ not_useful_field='blah')
```

(continues on next page)

(continued from previous page)

↳of the SCHEMA and is a

↳#
↳a
↳e
↳f
↳i
↳o
↳s
↳i
↳'s
↳n
↳a
↳p

↳#
↳n
↳f
↳i
↳t
↳c
↳f
my
↳n
↳p
↳=,
↳E
↳e
↳f
↳'
↳'

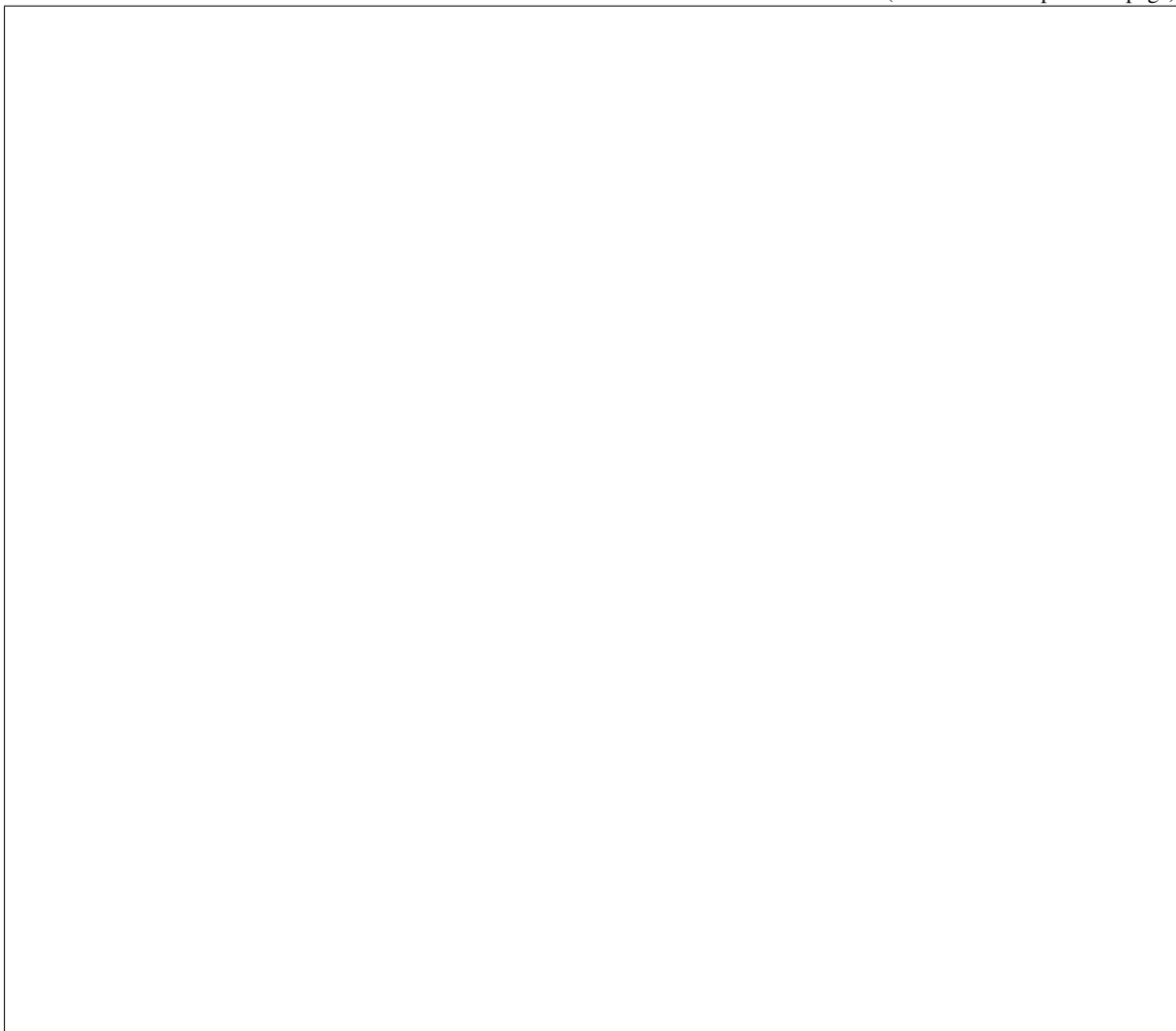
↳#
↳p
↳t
↳s
↳w
↳t
↳E
↳f
my_
↳n
↳p
↳p
↳s
↳o
↳e
↳o

required fields (`event_type`, `publisher_id`, and `level`, all sender fields needed by oslo that are defined in the ironic notification base classes) and emit it:

```
↪status=fields.NotificationStatus.START),
```

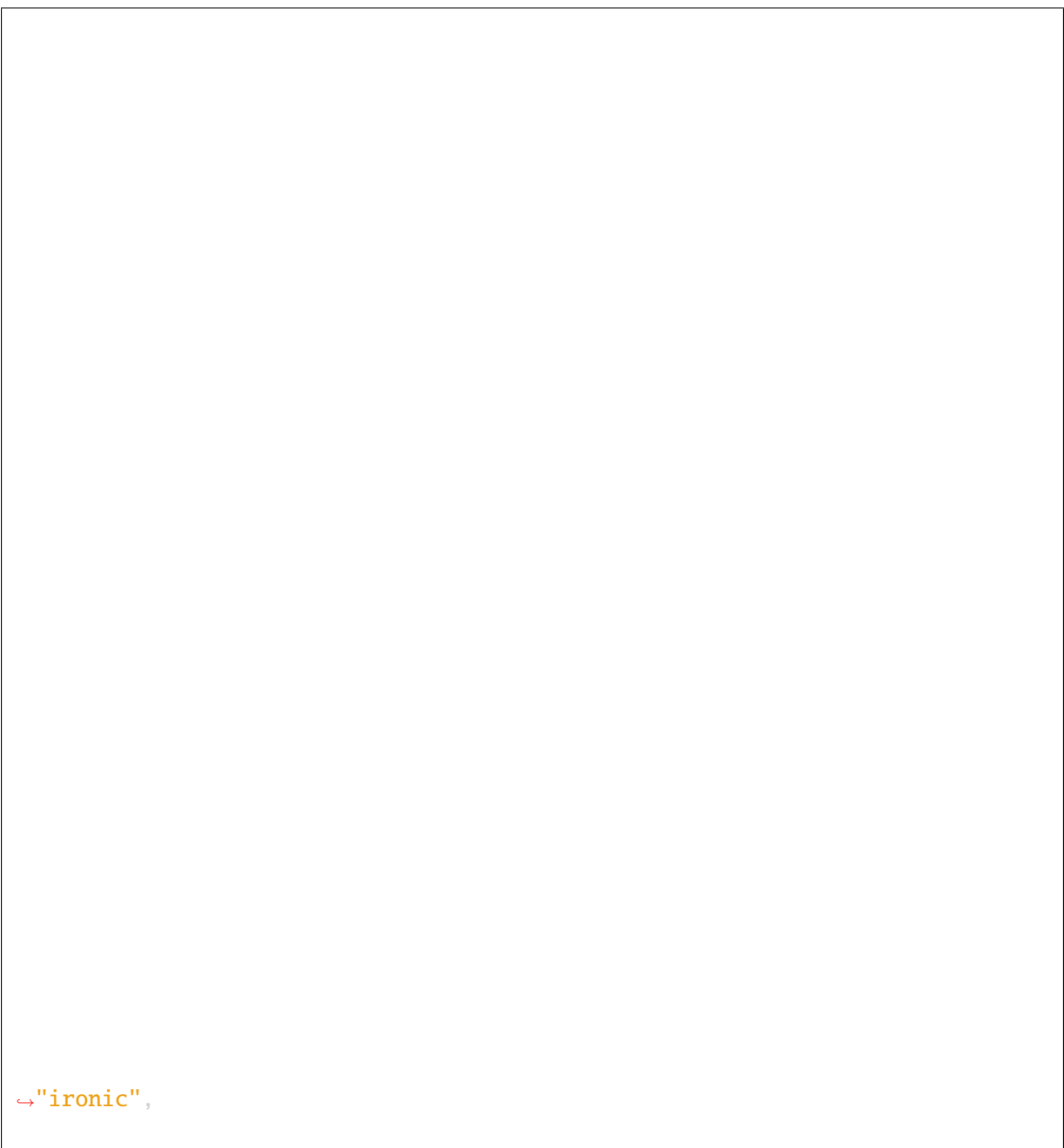
(continues on next page)

(continued from previous page)



ject being acted on, `action` will be a string describing what action is being performed on that object, and `status` will be one of start, end, error, or success. `start` and `end` are used to indicate when actions that are not immediate begin and succeed. `success` is used to indicate when actions that are immediate succeed. `error` is used to indicate when any type of action fails, regardless of whether its immediate or not. As a result of specifying these parameters, `event_type` will be formatted as `baremetal.<object>.<action>.<status>` on the message bus.

cation over the message bus:



(continues on next page)

(continued from previous page)

```
↪ "1.0",
↪
↪ {
↪
↪ "an_extra_field": "hello",
↪ "a_useful_field": "important",
```

(continues on next page)



About OSProfiler

brary. Its API provides different ways to add a new trace point. Trace points contain two messages (start and stop). Messages like below are sent to a collector:



(continues on next page)

OS-
Pro-
filer
is
an
Ope
Stac
cros
proj
pro-
fil-
ing
li-

→ {
└
→ └
→ └
→ └
→
→ "]
→ "
→
→ <
→ n
→ -
→ (
→
└
→ └
→ └
→ └
→
→ "
→ i
→ "
→
→ <
→ ,
→
└
→ └
→ └
→
→ "
→ i
→ "
→
→ <
→ ,
→
→

mation passed when calling profiler start() & stop() methods.

Two other alternatives for ceilometer are pure MongoDB driver and Elasticsearch.

-
<uui
of
par-
ent
trace
poin
tra
-
<uui
of
cur-
rent
trace
poin
inf
-
the
dic-
tio-
nary
that
con-
tains
user
in-
for-
The
pro-
filer
uses
ceil
ter
as
a
cen-
tral-
ized
col-
lec-
A
no-
ti-
fier

to ceilometer using oslo.messaging and ceilometer API is used to retrieve all messages related to one trace.

retrieve information about traces and present it in HTML/JSON using CLI.

library.

How to Use OSProfiler with Ironic in Devstack

is
setu
to
send
no-
ti-
fi-
ca-
tion

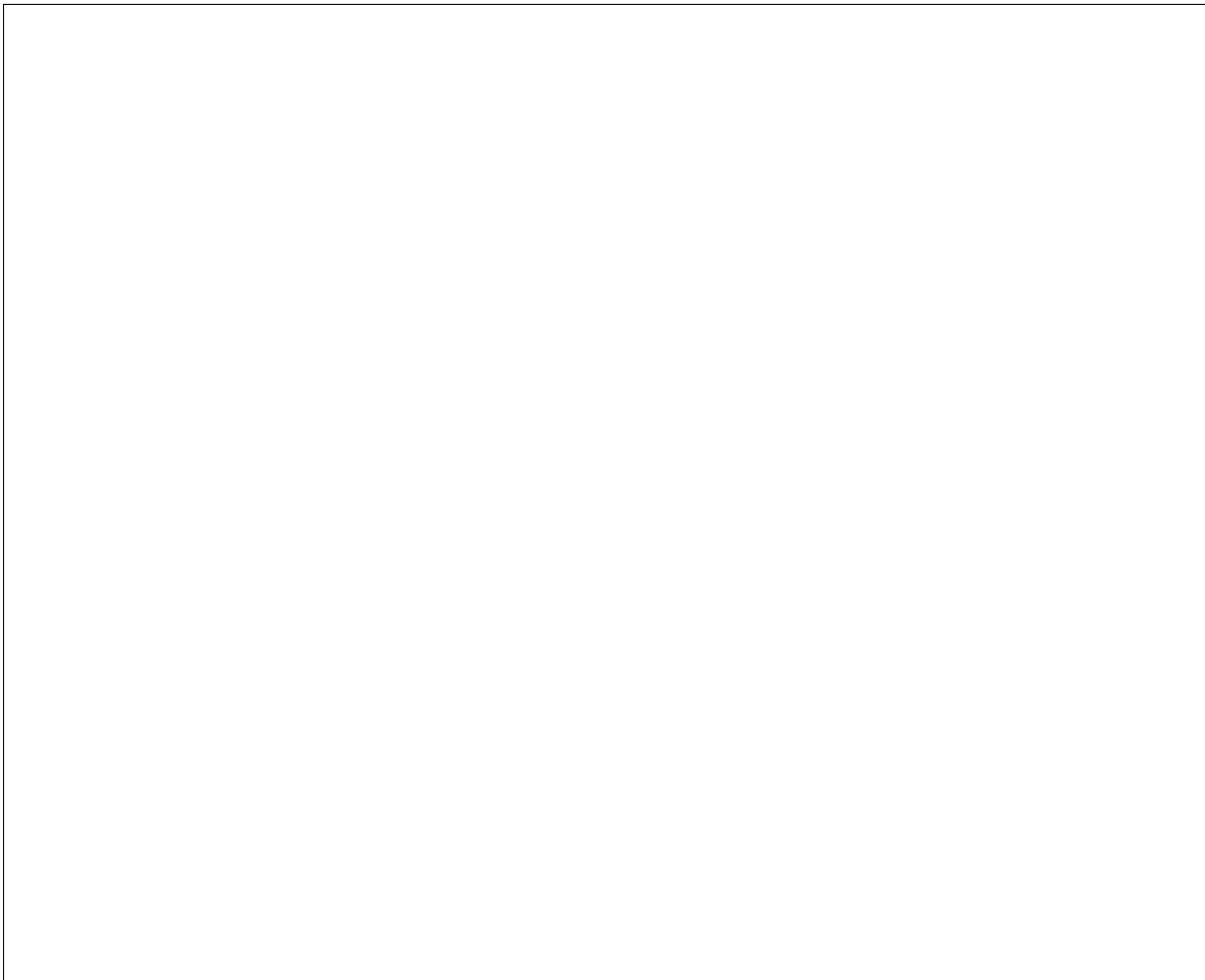
OS-
Pro-
filer
has
en-
try
poin
that
al-
lows
the
user
to

For
more
de-
tails
see
OS-
Pro-
filer
Cros
proj
pro-
fil-
ing

To
use
or
test
OS-
Pro-
filer

setup Devstack with OSProfiler and ceilometer. In addition to the setup described at *Deploying Ironic with DevStack*, the user needs to do the following:

and ceilometer:



(continues on next page)

in
iron
the
user
need
to

Add
the
fol-
low-
ing
to
loc
to
en-
able
OS-
Pro-
filer

enal
→p
→p
→h
→/
→o
→o
→o
→p
enal
→p
→c
→h
→
→o
→o
→o
→c
enal
→p
→o
→h
→/
→o
→o
→o

(continued from previous page)



→ #
→ E
→ t
→ f
→ s
CEI
→ N
→ T
→ p
ENAI
→ S
→ c
→ a
→ c
→ a
ENAI
→ S
→ c
→ a
→ c
→ c
ENAI
→ S
→ c
→ a
→ e
→ c
→ a
→ n
ENAI
→ S
→ c
→ a

Run
stack

Onc
De-
vs-
tack
en-
vi-
ron-
men-
is
setu
edit

to set the following profiler options and restart ironic services:

```
↪projects
```

stackclient to run baremetal commands with `--os-profile SECRET_KEY`.

iron
con.

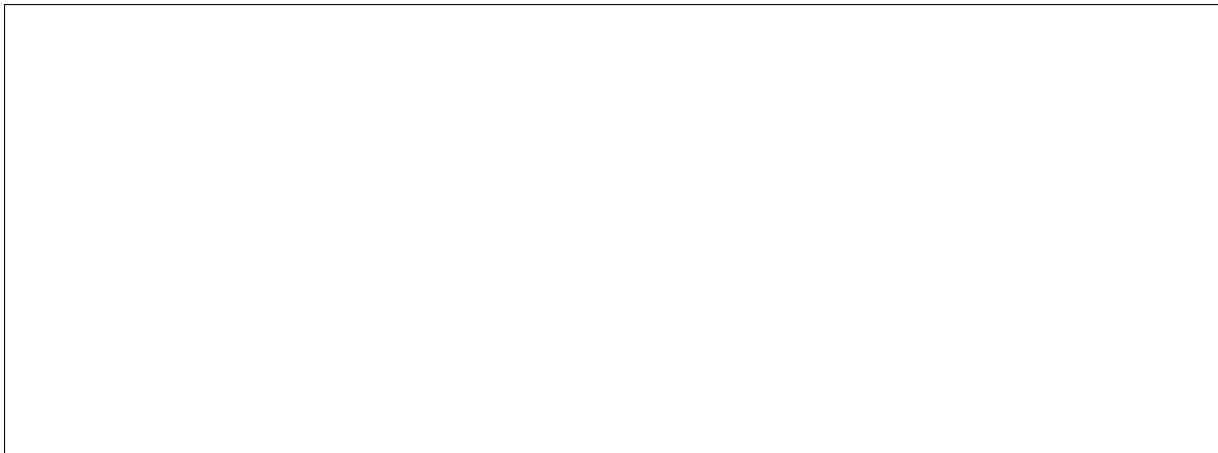
[pr
en
↪=
↪T
hma
↪k
↪=
↪S
↪K
↪
↪#
↪d
↪v
↪u
↪a
↪S
↪O

tra
↪S
↪=
↪T

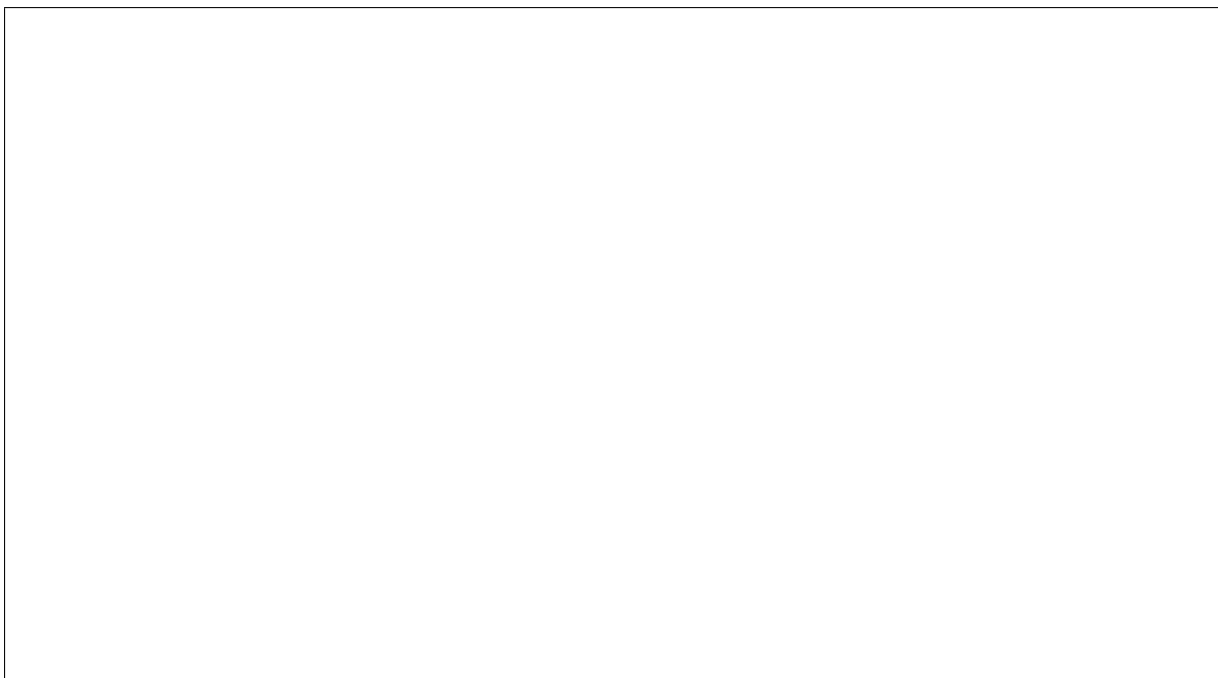
In
or-
der
to
trace
iron
us-
ing
OS-
Pro-
filer,
use
oper

For
ex-
am-
ple,
the
fol-
low-
ing
will
caus

to be printed after node list:



ing:



(continues on next page)

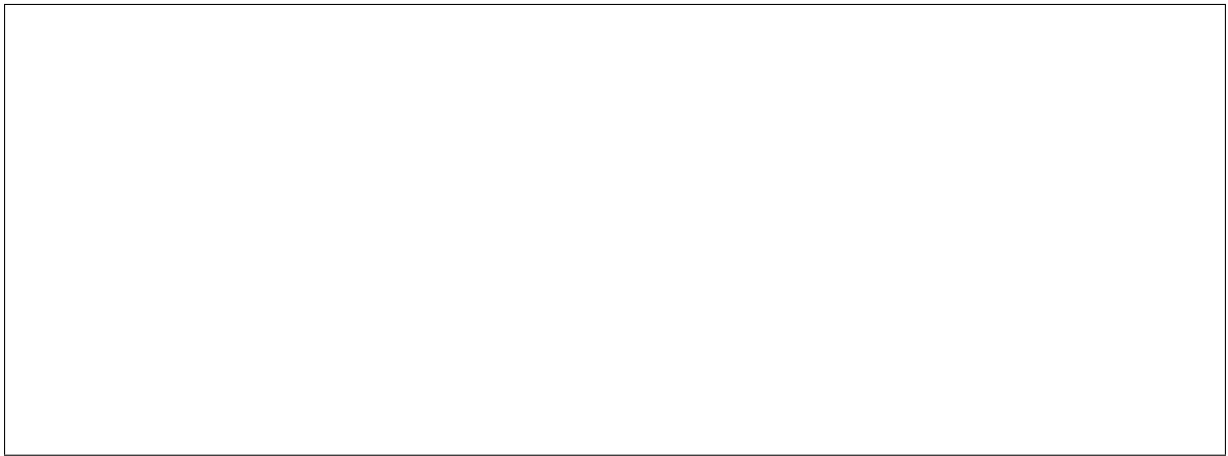
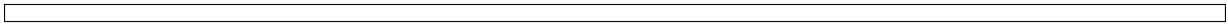
a
<tra
id>

↪ \$
↪ o
↪ -
↪ -
↪ o
↪ p
↪ S
↪ K
↪ b
↪ n
↪ l

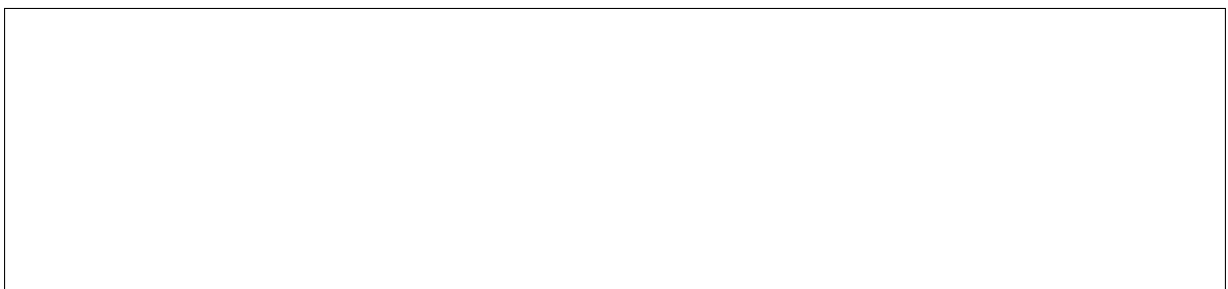
Out-
put
of
the
above
com
man
will
in-
clud
the
fol-
low-

Tra
↪ I
↪
↪ <
↪ i
↪
Disp
↪ t
↪ W
↪ c
osp
↪ t
↪ s
↪ -
↪ -
↪ h
↪
↪ <
↪ i
↪

(continued from previous page)



option:



(continues on next page)

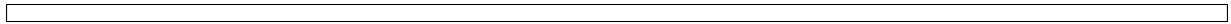
The
trace
re-
sults
can
be
seen
us-
ing
this
com
man

→ \$
→ o
→ t.
→ sl
→ -
→ -
→ h
→
→ <
→ i
→

The
trace
re-
sults
can
be
save
in
a
file
with
--o
fil

→ \$
→ o
→ t.
→ sl
→ -
→ -
→ h
→
→ <
→ i
→
→ -

(continued from previous page)



and db calls. More detailed db tracing is enabled if `trace_sqlalchemy` is set to true.

Levels	Duration	Type Project	Service	Host	Details
- 0	724 ms	total n/a	n/a	n/a	Details
1	9 ms	wsgi keystone	main	ubuntu	Details
1	311 ms	wsgi keystone	main	ubuntu	Details
- 1	367 ms	wsgi ironic	ironic_api	ubuntu	Details
2	81 ms	wsgi keystone	admin	ubuntu	Details
2	24 ms	db api ironic	ironic_api	ubuntu	Details
2	12 ms	db api ironic	ironic_api	ubuntu	Details
- 2	188 ms	rpc ironic	ironic_conductor	ubuntu	Details
3	35 ms	db api ironic	ironic_conductor	ubuntu	Details
3	41 ms	db api ironic	ironic_conductor	ubuntu	Details
3	14 ms	db api ironic	ironic_conductor	ubuntu	Details
3	8 ms	db api ironic	ironic_conductor	ubuntu	Details
3	41 ms	db api ironic	ironic_conductor	ubuntu	Details

low:

References

Details

- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)
- [Details](#)

The trace results show the time spent in ironic api, ironic con

Sample Trace

Each trace has embedded details as shown below

- OS-Profiler Cross-project profiling library
-

De-
ploy-
ing
Ironic
with
De-
vS-
tack

Rolling Upgrades

The
iron
iron
(iron
api
and
iron
conc
ser-
vice
sup-
port
rolli
up-
grad
start
ing
with
a
rolli
up-

grade from the Ocata to the Pike release. This describes the design of rolling upgrades, followed by notes for developing new features or modifying an IronicObject.

Design

Rolling upgrades between releases

Iron
fol-
lows
the
rele
cycl
with
inter
re-
lease
mod

`<major>.<minor>.<patch>`. We refer to a named release of ironic as the release associated with a development cycle like Pike.

deprecation period must be at least three months and a cycle boundary. This means that there will never be anything that is both deprecated *and* removed between two named releases.

The
re-
lease
are
sem
vers
in
the
form

In
ad-
di-
tion,
iron
fol-
lows
the
stan-
dard
dep-
re-
ca-
tion
pol-
icy,
whic
says
that
the

Roll
up-
grad
will
be
sup-
port
be-
twee

- nam
re-
lease
N
to
N+1
(star

cause those bug fixes can contain improvements to the upgrade process, the operator should patch the system before upgrading between named releases.

the above bullet point, there may be a bug or a feature introduced on a master branch, that we want to remove before publishing a named release. Deprecation policy allows to do this in a 3 month time frame. If the feature was included and removed in intermediate releases, there should be a release note

ing
with
N
==
Oca

•
any
nam
re-
leas
to
its
lat-
est
re-
vi-
sion
con-
tain-
ing
back
port
bug
fixes
Be-

•
mos
re-
cent
nam
re-
leas
N
(and
sem
re-
leas
new
than
N)
to
mas
ter.
As
with

added, with instructions on how to do a rolling upgrade to master from an affected release or release span. This would typically instruct the operator to upgrade to a particular intermediate release, before upgrading to master.

Rolling upgrade process

leases in this order:

Iron
sup-
port
rolli
up-
grad
as
de-
scrib
in
the
*up-
grad
guid*
The
up-
grad
pro-
cess
will
caus
the
iron
ser-
vice
to
be
run-
ning
the
From
and
ToV
re-
0.
Up-
grad
iron
code
and
run
data

sche
mi-
gra-
tions
via
the
iron
upg
com
man

1. Up-grad code and resta iron conc ser-vice one at a time

2. Up-grad code and resta iron api ser-vice one at a time

3. Un-pin API RPC and ob-ject ver-sion so that

est versions in ToVer. This is done via updating the configuration option described below in *API, RPC and object version pinning* and then restarting the services. ironic-conductor services should be restarted first, followed by the ironic-api services. This is to ensure that when new functionality is exposed on the unpinned API service (via API micro version), it is available on the backend.

step	ironic-api	ironic-conductor
0	all FromVer	all FromVer
1.1	all FromVer	some FromVer, some ToVer-pinned
1.2	all FromVer	all ToVer-pinned
2.1	some FromVer, some ToVer-pinned	all ToVer-pinned
2.2	all ToVer-pinned	all ToVer-pinned
3.1	all ToVer-pinned	some ToVer-pinned, some ToVer
3.2	all ToVer-pinned	all ToVer
3.3	some ToVer-pinned, some ToVer	all ToVer
3.4	all ToVer	all ToVer

Policy for changes to the DB model

to ironiC's [deprecation policy](#). But its alembic script has to wait one more deprecation period, otherwise an `unknown column` exception will be thrown when `FromVer` services access the DB. This is because **ironiC-dbsync upgrade** upgrades the DB schema but `FromVer` services still contain the dropped field in their SQLAlchemy DB model.

split it into multiple operations, with one operation per release cycle (to maintain compatibility with an old SQLAlchemy model). For example, to rename a column, add the new column in release N, then remove the old column in release N+1.

port
• The
drop
ping
of
colu
or
ta-
bles
and
cor-
re-
spor
ing
ob-
jects
field
is
sub-
ject

• An
aler
op.
alt
to
re-
nam
or
re-
size
a
col-
umn
is
not
al-
lowe
In-
stea

• Som

may impose table locks and cause downtime. If the change cannot be avoided and the impact is significant (e.g. the table can be frequently accessed and/or store a large dataset), these cases must be mentioned in the release notes.

API, RPC and object version pinning

a rolling upgrade, the services need to be able to handle different API, RPC and object versions.

im-
ple-
men-
ta-
tions
of
SQL
ALT
TAB
such
as
addi
for-
eign
keys
in
Post
greS

For
the
iron
ser-
vice
to
be
run-
ning
old
and
new
re-
lease
at
the
same
time
dur-
ing

This
ver-
sion
ing
is

used to pin the API, RPC and IronicObject (e.g., Node, Conductor, Chassis, Port, and Portgroup) versions for all the ironic services.

versions of API, RPC and IronicObjects. Its possible values are releases, named (e.g. ocata) or sem-versioned (e.g. 7.0).

han-
dled
via
the
con-
fig-
u-
ra-
tion
op-
tion:
[DE
pin.
It
is

The
de-
fault
valu
of
emp
in-
di-
cate
that
iron
api
and
iron
conc
will
use
the
lat-
est

In-
ter-
nally
in
com
mon
iron
main
tains
a
map

IronicObject versions associated with each release. This mapping is maintained manually.

uration option value to be the name (or version) of the old release. This will indicate to the services running the new release, which API, RPC and object versions that they should be compatible with, in order to communicate with the services using the old release.

Handling API versions

ping
that
in-
di-
cate
the
API
RPC
and

Dur-
ing
a
rolli
up-
grad
the
ser-
vice
us-
ing
the
new
re-
lease
will
set
the
con-
fig-

Whe
the
(new
ser-
vice
is
pinn
the
max
i-
mun
API
ver-
sion

pinned version which the older service supports (as described above at *API, RPC and object version pinning*). The ironic-api service returns HTTP status code 406 for any requests with API versions that are higher than this maximum version.

Handling RPC versions

and passes it to the `RPCClient` as an initialization parameter. This variable is then used to determine the maximum requested message version that the `RPCClient` can send.

it
sup-
port
will
be
the

Con
duc-
torA
sets
the
ver-
vari-
able
to
the
de-
sired
(lat-
est
or
pinn
RPC
API
ver-
sion

Eacl
RPC
call
can
cus-
tom
the
re-
ques
ac-
cord
ing
to
this
ver-
The

section below has more details about this.

Handling IronicObject versions

these boundaries, when the IronicObject enters or leaves the service, do we deal with object versioning:

*Ironic
RPC
ver-
sion*

In-
ter-
nally
iron
ser-
vice
deal
with
Ironic
i-
cOb
jects
in
their
lat-
est
ver-
sion
Only
at

- get-
ting
ob-
jects
from
the
data
con-
vert
to
lat-
est
ver-
sion
- sav-
ing
ob-
jects

version

latest version

to
the
data
if
pinn
save
in
pinn
ver-
sion
else
save
in
lat-
est

- se-
ri-
al-
iz-
ing
ob-
jects
(to
send
over
RPC
if
pinn
send
pinn
ver-
sion
else
send

- de-
se-
ri-
al-
iz-
ing
ob-
jects
(re-
ceiv
ing
ob-

est version

ture is supported by the API version and object versions. For example, when the ironic-api service is pinned, it can only allow actions that are available to the objects pinned version, and cannot allow actions that are only available for the latest version of that object.

jects
from
RPC
con-
vert
to
lat-

The
iron
api
ser-
vice
also
has
to
han-
dle
API
re-
ques
base
on
whe
or
how
a
fea-

To
sup-
port
this:

- All the data tables (SQL models) of the Iron i-cOb jects

version. The value is the version of the object that is saved in the database.

turned. Otherwise, the latest version is returned.

sion may be a newer or older version than the existing version of the object. The bulk of the work is done in the helper method `IronicObject._convert_to_version()`. Subclasses that have new versions redefine this to perform the actual conversions.

have
a
col-
umn
nam

- The
meth
Iron
get.
re-
turn
the
tar-
get
ver-
sion
If
pinn
the
pinn
ver-
sion
is
re-

- The
meth
Iron
con
con-
vert
the
ob-
ject
into
the
tar-
get
ver-
sion
The
tar-
get
ver-

In the following,

- The old release is From it uses version 1.14 of a Node object.

- The new release is ToV It uses version 1.15 of a Node object this has a

deprecated `extra` field and a new `meta` field that replaces `extra`.

- `db_c` and `db_c` are the data

Getting objects from the database (API/conductor < DB)

to IronicObjects via the method `IronicObject._from_db_object()`. This method always returns the IronicObject in its latest version, even if it was in an older version in the database. This is done regardless of the service being pinned or not.

rep-
re-
sen-
ta-
tions
of
thos
node
field

Both
iron
api
and
iron
conc
ser-
vice
read
val-
ues
from
the
data
The
val-
ues
are
con-
verte

Note
that
if
an
ob-
ject
is
con-
verte
to
a
later
ver-
sion
that

retain any changes (in its `_changed_fields` field) resulting from that conversion. This is needed in case the object gets saved later, in the latest version.

it doesnt exist)

Iron
i-
cOb
ject
will

For
ex-
am-
ple,
if
the
node
in
the
data
is
in
ver-
sion
1.14
and
has
db_c
set:

- a
From
ser-
vice
will
get
a
Node
with
node
=
db_c
(and
no
know
edge
of
node
since

- a
ToV

ser-
vice
(pin
or
un-
pinn
will
get
a
Nod
with

-
node
=
db_c

-
node
=
Non

-
node
=
[met
ex-
tra]

Saving objects to the database (API/conductor > DB)

The
ver-
sion
used
for
sav-
ing
Iron
i-
cOb
jects
to
the
data
is
de-
ter-
min
as
fol-

lows:

jects are always in their latest version, no conversions are needed.

always in their latest version, the object needs to be converted to the pinned version before being saved.

- For an unpinned service the object is saved in its latest version. Since ob-

- For a pinned service the object is saved in its pinned version. Since objects are

The method `Iron.do_` handles this logic

new values (similar to the existing `oslo.versionedobjects.VersionedObject.obj_get_changes()`). Since we do not keep track internally, of the database version of an object, the objects `version` field will always be part of these changes.

saved in its latest version, all services are running the newer release (although some may still be pinned) and can handle the latest object versions.

re-
turn
ing
a
dic-
tio-
nary
of
char
field
and
their

The
*Roll
up-
graa
pro-
cess*
(at
step
3.1)
en-
sure
that
by
the
time
an
ob-
ject
can
be

An
in-
ter-
est-
ing
sit-
u-
a-
tion
can
oc-
cur
whe

in step 3.1. It is possible for an `IronicObject` to be saved in a newer version and subsequently get saved in an older version. For example, a `ToVer` unpinned conductor might save a node in version 1.5. A subsequent request may cause a `ToVer` pinned conductor to replace and save the same node in version 1.4!

Sending objects via RPC (API/conductor -> RPC)

that request are serialized into entities or primitives via `IronicObjectSerializer.serialize_entity()`. The version used for objects being serialized is as follows:

the
ser-
vice
are
as
de-
scrib

When
a
ser-
vice
mak
an
RPC
re-
ques
any
Iron
i-
cOb
jects
that
are
sent
as
part
of

- For
an
un-
pinn
ser-
vice
the
ob-
ject
is
se-
ri-
al-

sion. Since objects are always in their latest version, no conversions are needed.

objects are always in their latest version, the object is converted to the pinned version before being serialized. The converted object includes changes that resulted from the conversion; this is needed so that the service at the other end of the RPC request has the necessary information if that object will be saved to the database.

Receiving objects via RPC (API/conductor <- RPC)

ized
to
its
lat-
est
ver-

- For a pinned service the object is serialized to its pinned version. Since

When a service receives an RPC request any entities that are

quest need to be deserialized (via `oslo.versionedobjects.VersionedObjectSerializer.deserialize_entity()`). For entities that represent `IronicObjects`, we want the deserialization process (via `IronicObjectSerializer._process_object()`) to result in `IronicObjects` that are in their latest version, regardless of the version they were sent in and regardless of whether the receiving service is pinned or not. Again, any objects that are converted will retain the changes that resulted from the conversion, useful if that object is later saved to the database.

version 1.4, where `node.extra` was changed (so `node._changed_fields = [extra]`). This node will be serialized in version 1.4. The receiving `ToVer` pinned `ironic-conductor` deserializes it and converts it to version 1.5. The resulting node will have `node.meta` set (to the changed value from `node.extra` in v1.4), `node.extra = None`, and `node._changed_fields = [meta, extra]`.

When developing a new feature or modifying an `IronicObject`

that things work during a rolling upgrade.

points to keep in mind when developing code.

ironic-api

ject,
they
need
to
be
code
so

The
fol-
low-
ing
de-
scrib
ar-
eas
whe
the
code
may
need
to
be
char
as
well
as
som

Dur-
ing
a
rolli
up-
grad
the
new
pinn
iron
api
is
talk-
ing
to
a
new
con-

that might also be pinned. There may also be old ironic-api services. So the new, pinned ironic-api service needs to act like it was the older service:

in the old and new releases. Pinning the API version is in place to handle this.

being handled that cannot or should not be handled, it should be coded so that the response has HTTP status code 406 (Not Acceptable). This is the same response to requests that have an incorrect (old) version specified.

duc-
tor

- New
fea-
ture
shou
not
be
mad
avai
able
un-
less
they
are
som
how
to-
tally
sup-
port

—
If,
for
wha
ever
rea-
son,
the
API
ver-
sion
pin-
ning
does
pre-
vent
a
re-
ques
from

IroniC RPC versions

the following needs to be considered:

conductor/rpcapi.py, used by ironic-api) and the server (ironic/conductor/manager.py, used by ironic-conductor). It should also be updated in ironic/common/release_mappings.py.

When
the
sig-
na-
ture
(ar-
gu-
men-
of
an
RPC
meth-
is
chan-
or
new
meth-
ods
are
add-

- The
RPC
ver-
sion
must
be
in-
cre-
men-
and
be
the
sam-
valu-
for
both
the
clien-
(irc

- Un-
til
there
is

only be added as optional. Existing arguments cannot be removed or changed in incompatible ways with the method in older RPC versions.

structor of `oslo_messaging.RPCClient`). This pinning is in place during a rolling upgrade when the `[DEFAULT]/pin_release_version` configuration option is set.

a
ma-
jor
ver-
sion
bum
new
ar-
gu-
men
of
an
RPC
meth
can

- iron
api
(clie
side
sets
a
ver-
sion
cap
(by
pass
ing
the
ver-
sion
cap
to
the
con-

- New
RPC
meth
ods
are
not
avai
able
whe
the

version. In this case, the corresponding REST API function should return a server error or implement alternative behaviours.

request is compatible with the version cap of the RPC Client. Otherwise the request needs to be created to work with a previous version that is supported.

ser-
vice
is
pinn
to
the
olde
re-
leas

- Met
ods
whic
char
ar-
gu-
men
shou
run
cli
can
to
see
if
the
ver-
sion
of
the

- iron
conc
(serv
side
shou
tol-
er-
ate
olde
ver-
sion
of
re-
ques
in
or-

working during the rolling upgrade process. The behaviour of ironic-conductor will depend on the input parameters passed from the client-side.

ous named release.

Object versions

sidered:

der
to
keep

- Old methods can be removed only after they are no longer used by a previous

When subclassed of ironic objects based on Iron are modified, the following need to be considered:

the object version. The object versions are also maintained in `ironic/common/release_mappings.py`.

be excluded from the version check by adding their class names to the `NEW_MODELS` list in `ironic/cmd/dbsync.py`.

- Any character of field or character in signature of remote methods need a bunch of
- New objects must be added to `ironic/common/release_mappings.py`. Also for the first release they should
- The argument of

tor via RPC) can only be added as optional. They cannot be removed or changed in an incompatible way (to the previous release).

one.

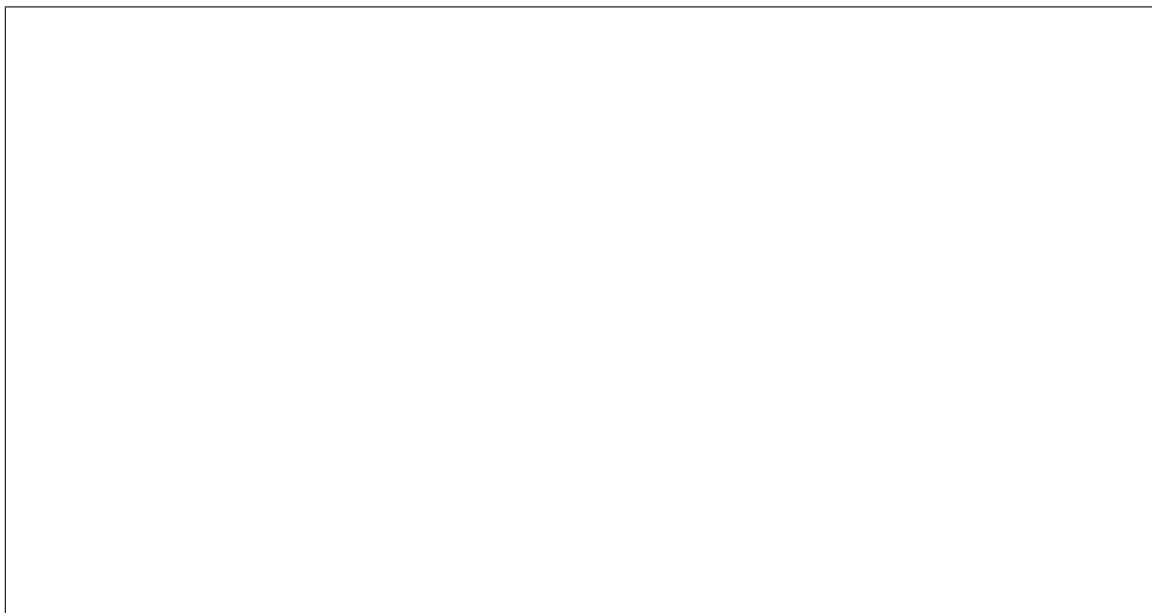
re-
mota
meth
ods
ods
(met
ods
whic
are
re-
mote
to
the
con-
duc-

- Field type cannot be changed. Instead, create a new field and deprecate the old

- There is a unit test that generates the hash of

and the signatures of its remotable methods. Objects that have a version bump need to be updated in the `expected_object_fingerprints` dictionary; otherwise this test will fail. A failed test can also indicate to the developer that their change(s) to an object require a version bump.

ing or writing to the database, `ironic.objects.base.IronicObject._convert_to_version()` will be called to convert objects to the target version. Objects should implement their own `._convert_to_version()` to remove or alter fields which were added or changed after the target version:

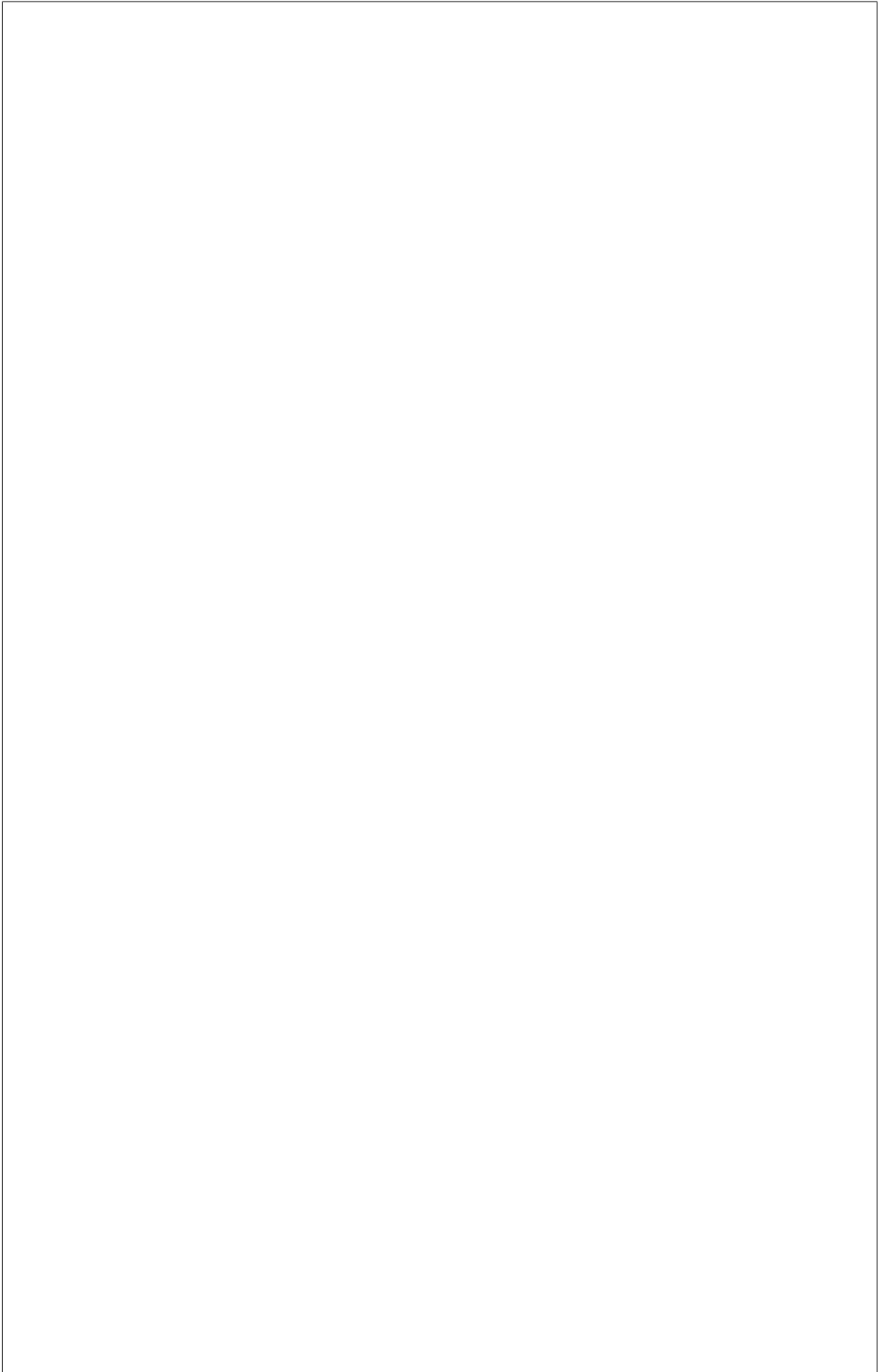


(continues on next page)

an
ob-
ject
us-
ing
its
field

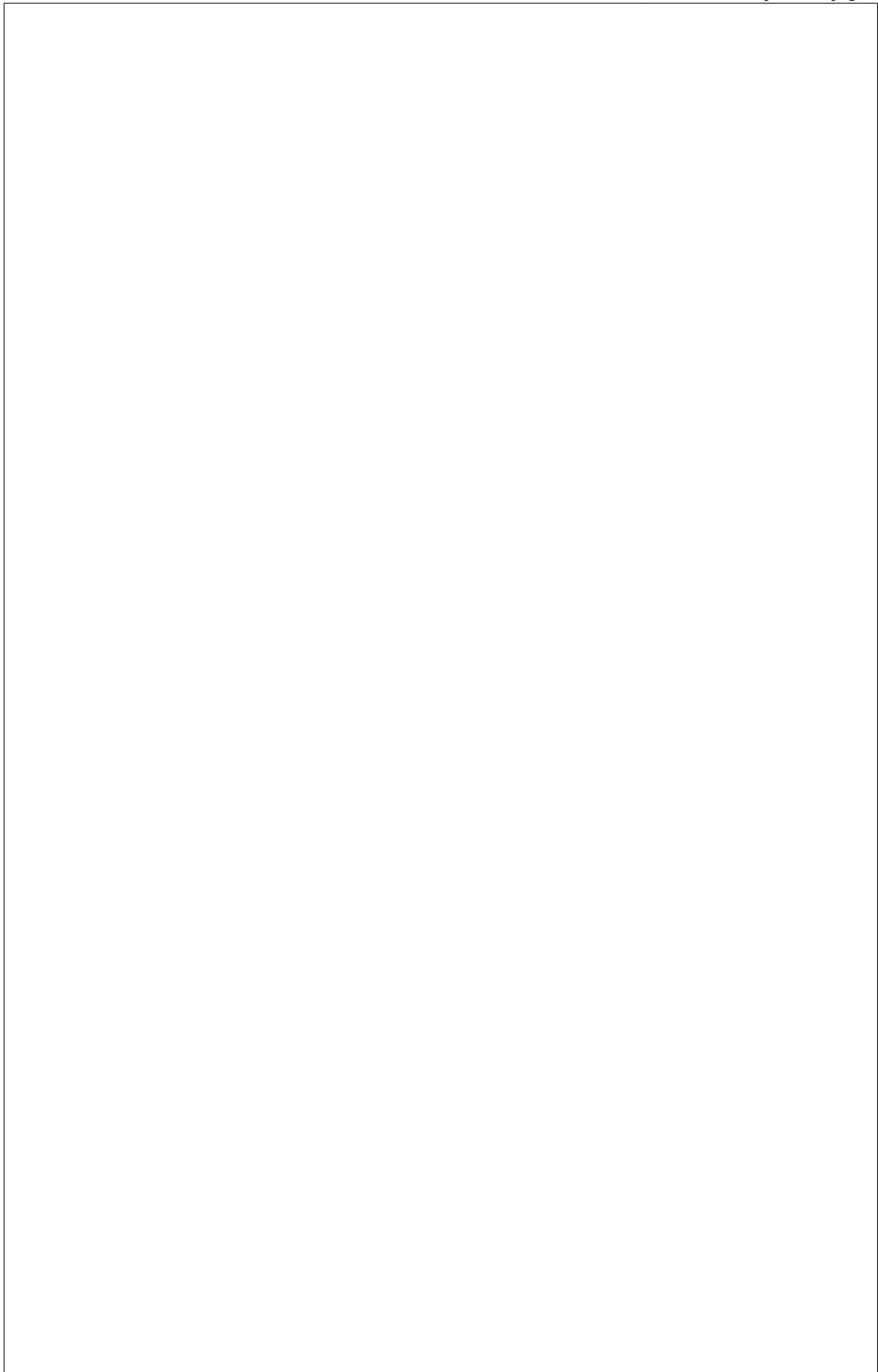
•
Whe
new
ver-
sion
ob-
jects
com
mu-
ni-
cate
with
old
ver-
sion
ob-
jects
and
whe
read

(continued from previous page)



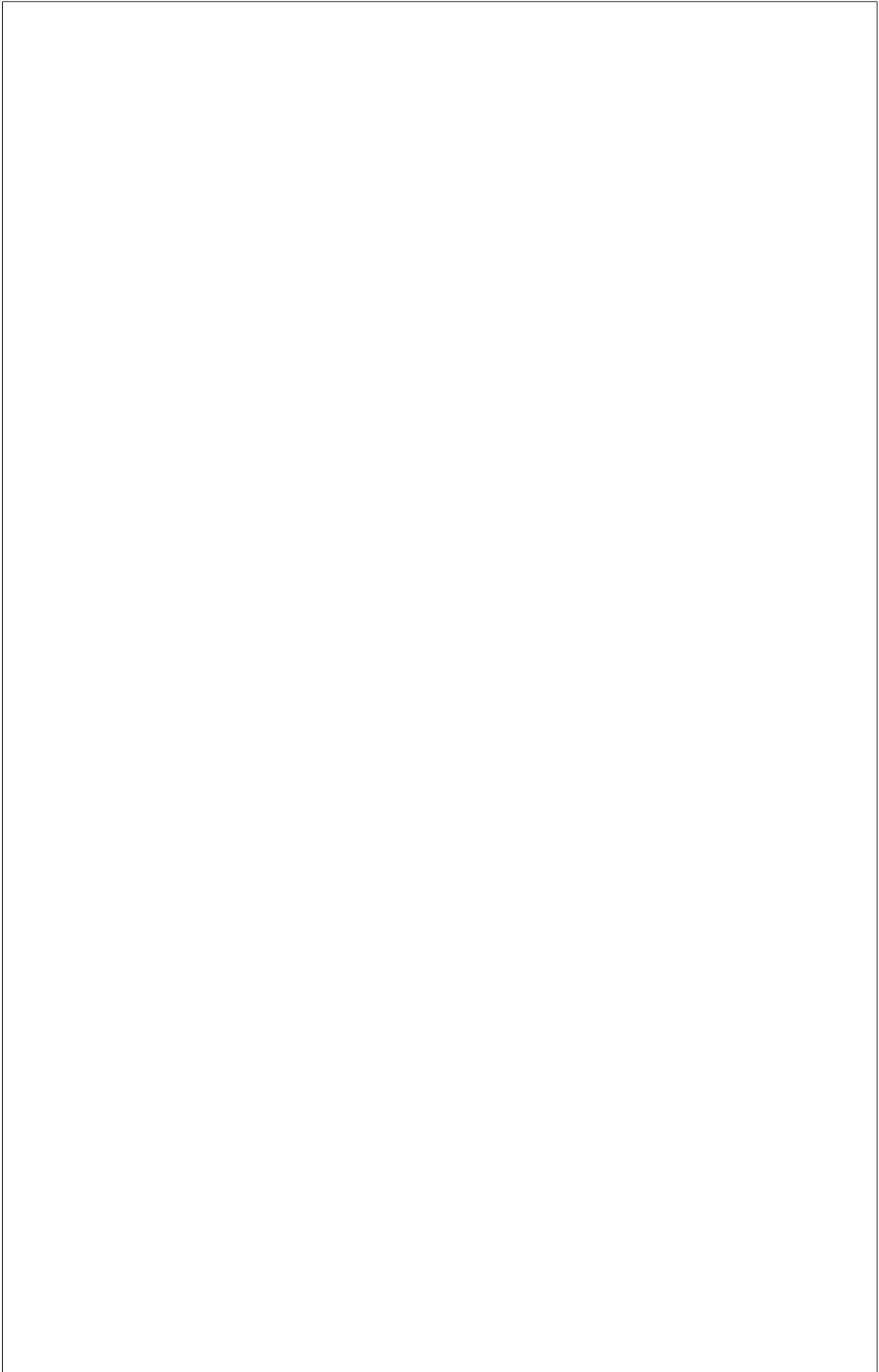
(continues on next page)

(continued from previous page)



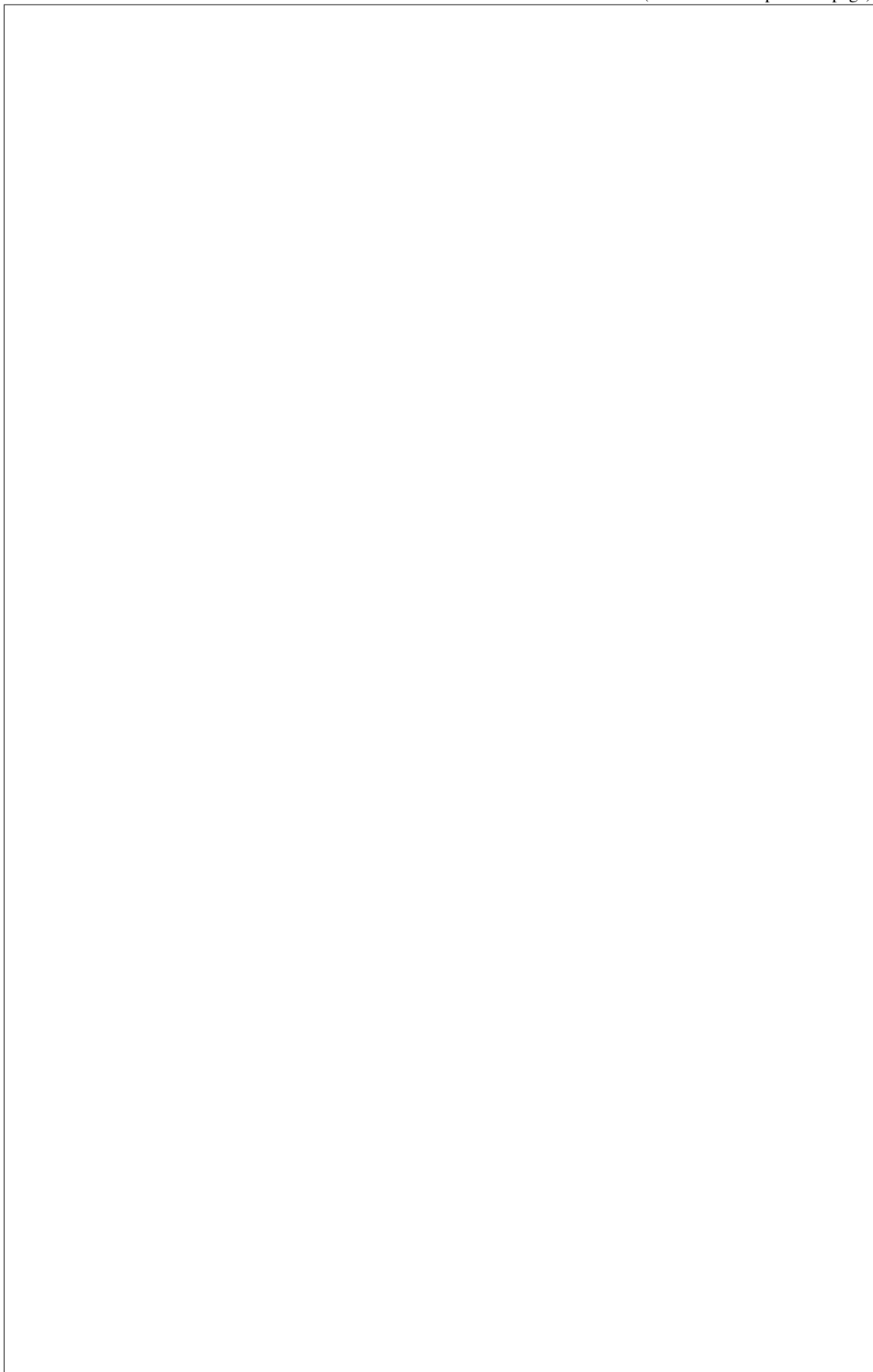
(continues on next page)

(continued from previous page)



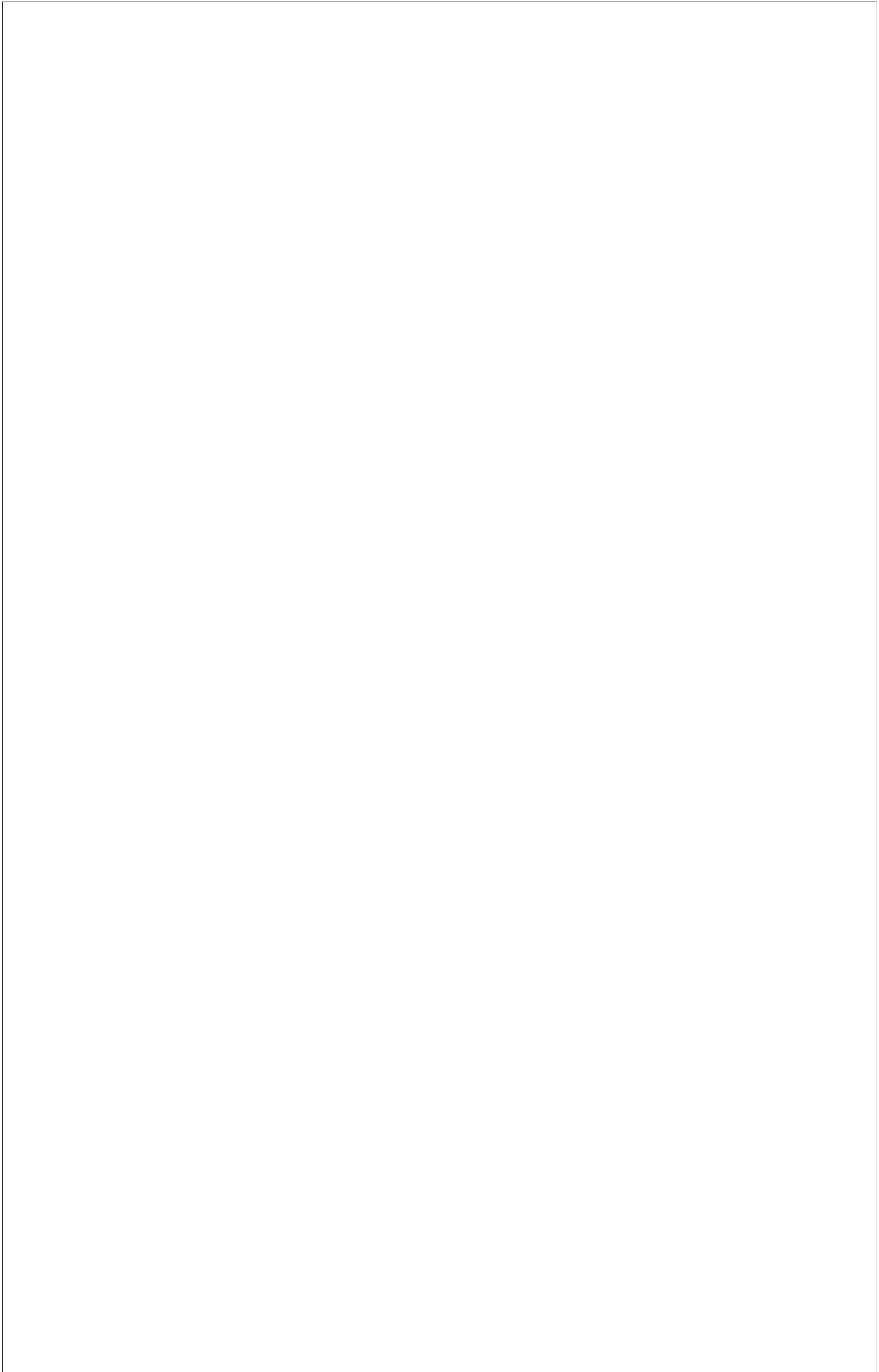
(continues on next page)

(continued from previous page)



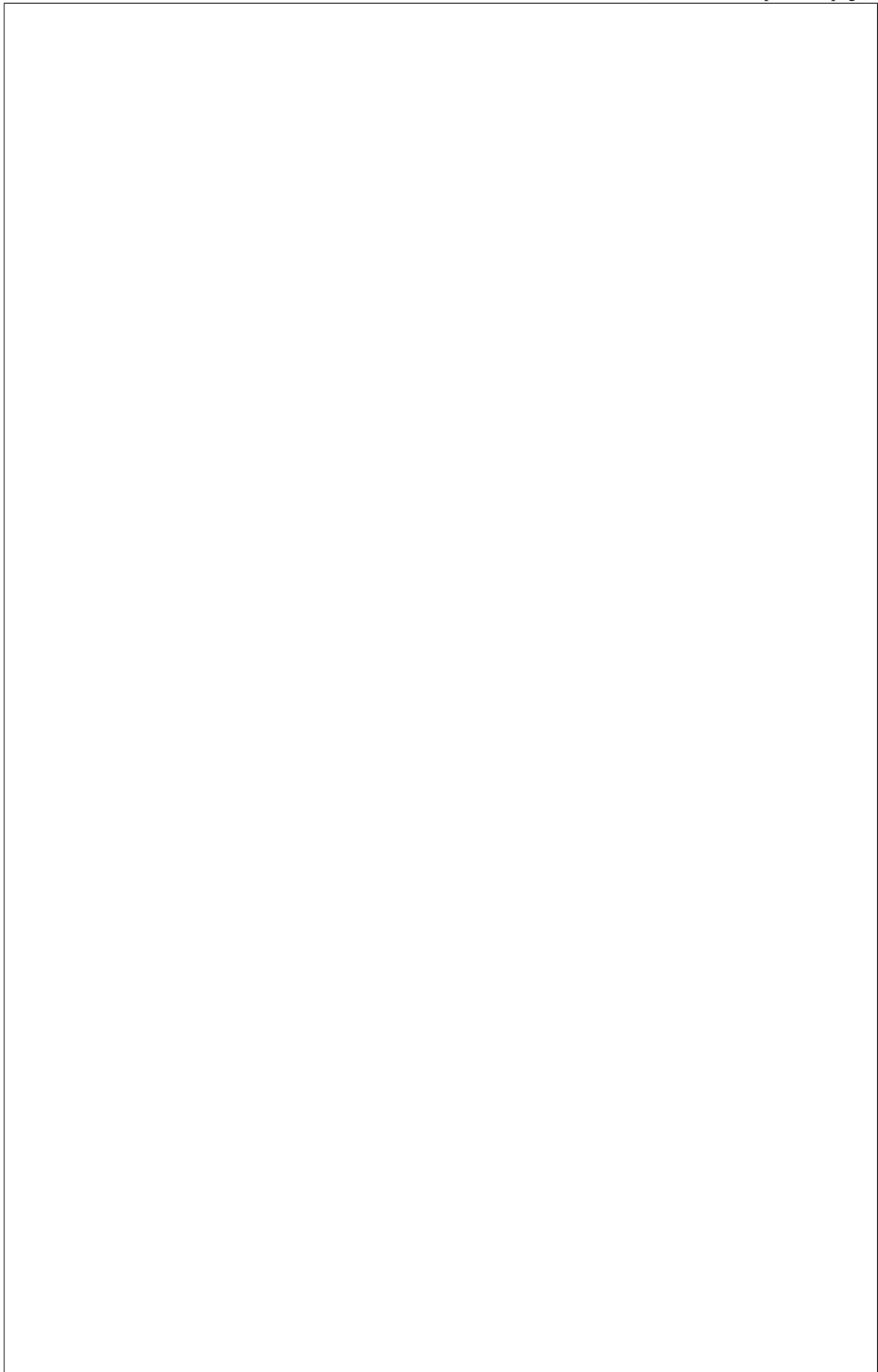
(continues on next page)

(continued from previous page)



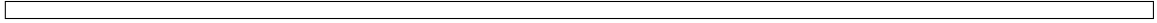
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



This
meth
mus
han-
dle:

-
con-
vert-
ing
from
an
olde
ver-
sion
to
a
new
ver-
sion

-
con-
vert-
ing
from
a
new
ver-
sion
to
an
olde
ver-
sion

-
mak
ing
sure
whe
con-
vert-
ing,
that
you
take
into
con-
sid-

fields that may have been affected by a field (value) only available in a newer version. For example, if field `new` is only available in Node version 1.5 and `Node.affected = Node.new+3`, when converting to 1.4 (an older version), you may need to change the value of `Node.affected` too.

Online data migrations

in SQLAlchemy models, like removing or renaming columns and tables can break rolling upgrades (when ironic services are run with different release versions simultaneously). It is forbidden to remove these database resources when they may still be used by the previous named release.

er-
a-
tion
othe
ob-
ject

The
iron
onl.
com
man
will
per-
form
on-
line
data
mi-
gra-
tions

Keep
in
mind
the
*Pol-
icy
for
char
to
the
DB
mod*
Fu-
ture
in-
com
pat-
i-
ble
char

that any new columns default to NULL. Test the migration out on a non-empty database to make sure that any new constraints dont cause the database to be locked out for normal operations.

[mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html](https://dev.mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html). (You should also check older, widely deployed InnoDB versions for issues.) In the case of PostgreSQL, adding a foreign key may lock a whole table for writes.

When
cre-
at-
ing
new
Alter
bic
mi-
gra-
tions
which
mod
ify
ex-
ist-
ing
mod
els,
mak
sure

You
can
find
an
over
on
what
DDL
op-
er-
a-
tions
may
caus
dow
time
in
[https://dev.](https://dev.mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html)

Mak
sure
to
add
a

must be implemented inside an online migration script. A script is a database API method (added to `ironic/db/api.py` and `ironic/db/sqlalchemy/api.py`) which takes two arguments:

re-
lease
note
if
there
are
any
dow
relat
con-
cern

Back
fill-
ing
de-
fault
val-
ues,
and
mi-
grat-
ing
data
be-
twee
colu
or
be-
twee
ta-
bles

•
con-
text:
an
ad-
min
con-
text

•
max
this
is
used
to
limi
the
quer

to migrate; ≥ 0 . If zero, all the objects will be migrated.

method, and

It
is
the
max
i-
mun
num
ber
of
ob-
jects

It
re-
turn
a
two-
tuple

- the
to-
tal
num
ber
of
ob-
jects
that
need
to
be
mi-
grate
at
the
start
of
the

- the
num
ber
of
mi-
grate
ob-
jects

In

this
meth
the
ver-
sion
col-
umn
can
be
used
to
se-
lect
and
up-
date
old
ob-
jects

The
meth
nam
shou
be
adde
to
the
list
of
ONL
in
iron
cmd,
dbs
py.

The
meth
shou
be
re-
mov
in
the
next
nam
re-
leas
af-
ter
this

old fields, old columns can be removed from the database. This takes at least 3 releases, since we have to wait until the previous named release no longer contains references to the old schema. Before removing any resources from the database by modifying the schema, make sure that your implementation checks that all objects in the affected tables have been migrated. This check can be implemented using the version column.

ironic-dbsync upgrade command

with the (new) release of ironic, before it will make any DB schema changes. If one or more objects are

one.
Af-
ter
on-
line
data
mi-
gra-
tions
are
com-
plete
and
the
SQL
mod-
els
no
long
con-
tain

The
ironic
upg-
com-
man-
first
check
that
the
ver-
sion
of
the
ob-
jects
are
com-
pat-
i-
ble

not compatible, the upgrade will not be performed.

(or supported) versions of these objects. The supported versions are the versions specified in `ironic.common.release_mappings.RELEASE_MAPPING`. The newly created tables cannot pass this check and thus have to be excluded by adding their object class names (e.g. `Node`) to `ironic.cmd.dbsync.NEW_MODELS`.

Role Based Access Control - Testing

standard pattern of entirely python based unit testing. In part this was done for purposes of speed and to keep the declaration of the test context.

This
check
is
done
by
com
par-
ing
the
ob-
jects
ver-
field
in
the
data
with
the
ex-
pect

The
Role
Base
Ac-
cess
con-
trol
test-
ing
is
a
mi-
nor
de-
par-
ture
from
the
Iron
stan-

is required to properly migrate the Ironic project from a project scoped universe where an admin project is utilized as the authenticating factor coupled with two custom roles, `baremetal_admin`, and `baremetal_observer`.

tional tests in place using this method, it definitely helped the speed at which these were created, and then ported to support additional.

This
also
lend
it-
self
to
be
very
use-
ful
due
to
the
na-
ture
of
A/B
test-
ing
whic

As
a
con-
trib-
u-
tor
look
ing
back
af-
ter
get-
ting
a
over
a
thou
sand
ad-
di-

How these tests work

header, which settings to prevent the `keystonemiddleware` from intercepting and replacing the headers were passing. Ultimately this is a feature, and it helps quite a bit.

`get_random_topic_for` methods. These calls raise `Temporary Unavailable`, since trying to execute the entire interaction into the conductor is moderately pointless because all policy enforcement is located with-in the API layer.

The
tests
ex-
e-
cute
API
calls
thro
the
API
layer
us-
ing
the
ap-
pro-
pri-
ate
verb
and

The
sec-
ond
as-
pect
of
how
this
worl
is
were
moc
ing
the
con-
duc-
tor
RPC
get.
and

At
the
sam

would have been a heavier lift. As such, the tests largely look for one of the following error codes.

tion.

time
wiri
ev-
ery-
thing
up
to
go
from
API
to
con-
duc-
tor
code
and
back

- 200
-

Got
the
item
from
the
API
-

This
is
an
data
driv
in-
ter-
ac-

- 201
-

Cre-
ated
-

This
is
data

drive
in-
ter-
ac-
tion.
These
are
rare.

- 204 -

Ac-
cept
-

This
is
a
data
drive
in-
ter-
ac-
tion.
These
are
rare.

- **403 - Forbidden**
ac-
cess
was
de-
nied

- **404 - Not Found**
Iron
be-
com-
scop
awa
thes
are
gen-
er-
ally
only
in
the
drive
API

ior. In System scope aware Project scoped configuration, i.e. later RBAC tests, this will become the dominant response for project scoped users as responding with a 403 if they could be an owner or lessee would provide insight into the existence of a node.

to the conductor.

How to make changes or review these tests?

end-
poin
be-
hav-

• 503 - Ser
whe
a
re-
ques
has
been
suc-
cess
fully
au-
then
ti-
cate
and
wou
have
been
sent
alon

The
tests
cy-
cle
thro
the
var-
i-
ous
end-
poin
and
re-
peat
ing
pat-
terns
are
clear

ible. Typically this means a given endpoint is cycled through with the same basic test using slightly different parameters such as different authentication parameters. When it comes to system scope aware tests supporting `node owners` and `lessee`, these tests will cycle a little more with slightly different attributes as the operation is not general against a shared common node, but different nodes.

returned. This is important later with `owner` and `lessee` having slightly different views of the universe.

vis-

Som
tests
will
test
body
con-
tents
or
at-
tribu
som
will
val-
i-
date
the
num
ber
of
reco

Som
gen-
eral
rules
ap-
ply

- Ad-
min
can
do
thing
at
least
as
far
as
their
scop
or
righ
ap-
ply.
Re-

owner and lessee admins are closer to System scoped Admin Members.

fields containing infrastructure internal addresses, these values will become hidden and additional tests will examine this.

men
ber:

• Men
bers
can
do
som
thing
but
not
ev-
ery-
thing

• Rea
ers
can
al-
way
read
but
as
we
get
into
sen-
si-
tive
data
later
on
such
as

• Thir
part
or
ex-
ter-
nal/
Ad-
mins
will
find
noth
ing

404, or even 500 errors.

What is/will be tested?

Access Control related capabilities will come in a series of phases, styles vary a little.

but
sad-
ness
in
emp
lists.
403,

The
idea
is
to
in
esse
test
as
muc
as
pos-
si-
ble,
how
ever
as
thes
tests
Role
Base

The
first
phas
is
"le
In
esse
thes
are
par-
tially
pro-
gra-
mat-
i-
cally
gen-
er-

then human reviewed and values populated with expected values.

execute the legacy tests with the updated `oslo.policy` configuration to help enforce scopes. These tests will intentionally begin to fail in phase three.

this process, as various portions of the API are made system scope aware. The legacy tests are marked as `deprecated` which signals to the second phase test sequences that they are **expected** to fail. New `system` scoped tests are also implemented which are matched up by name to the legacy tests. The major difference being some header values, and a user with a `member` role in the `system` scope now has some rights.

ated
and

The
sec-
ond
phas
is
re-
marl
ably
sim-
i-
lar
to
leg
It
is
the
safet
net
whe
we

The
thir
phas
is
the
im-
ple-
men
ta-
tion
of
Sys-
tem
scop
awar
ness
for
the
API
In

proach is similar, however it is much more of a shotgun approach. We test what we know should work, and what know should not work, but we do not have redundant testing for each role as `admin` users are also `members`, and since the policy rules are designed around thresholds of access, it just made no sense to run the same test for admin and members, where member was the threshold. These thresholds will vary with the proposed default policy. The forth scope also tests a third party external admin as a negative test to ensure that we are also denying access to resources appropriately.

The
forth
phas
is
im-
ple-
men
tatio
of
own
and
les
awa
proj
scop
ing.
The
test-
ing
ap-

The
page
con-
tain
in-
for-
ma-
tion
for
PTL
cros
proj
li-
aiso
and
core
re-
view
ers.

Releasing Ironic Projects

that process here.

Since
the
re-
spor
si-
bil-
ity
for
re-
lease
will
mov
be-
twee
peo-
ple,
we
doc-
u-
men

A
full
list
of
proj
that
iron
man
ages
is
avai
able
in
the
gov-
er-
nan
site.

Who is responsible for releases?

They may choose to delegate this responsibility to a liaison, which is documented in the [cross-project liaison wiki](#).

liaison must +1 the request for it to be processed.

The
cur-
rent
PTL
is
ul-
ti-
mate
re-
spor-
si-
ble
for
mak-
ing
sure
code
gets
re-
leas-

Any
one
may
sub-
mit
a
re-
leas-
re-
ques-
per
the
pro-
cess
be-
low,
but
the
PTL
or

Release process

mented in the [Project Team Guide](#).

What do we have to release?

itimate source of truth for this is [projects.yaml](#) in the governance repository. These deliverables have varying release models, and these are defined in the [deliverables YAML files](#) in the releases repository.

Re-
lease
are
man
aged
by
the
Ope
Stac
re-
lease
team
The
re-
lease
pro-
cess
is
doc-
u-

The
iron
proj
has
a
num
ber
of
de-
liv-
er-
able
un-
der
its
gov-
er-
nanc
The
ul-

In
gen-

Non-client libraries

eral,
iron
de-
liv-
er-
able
fol-
low
the
cycl
with
inter
re-
lease
mod

The
fol-
low-
ing
de-
liv-
er-
able
are
non-
clien
li-
brar

- iron
lib
- met-
al-
smit
- sush

Client libraries

The following deliverables are client libraries

- python-ironic
- python-ironic-inspector-client

Normal release

The following deliverables are Neutron plugins:

- neutron-baremetal
- neutron-swift

The following

low-
ing
de-
liv-
er-
able
are
Hor
zon
plu-
g-
ins:

- iron
ui

The
fol-
low-
ing
de-
liv-
er-
able
are
Tem
pest
plu-
g-
ins:

- iron
temp
plug

The
fol-
low-
ing
de-
liv-
er-
able
are
ser-
vice
or
treat
as
such

-

bifro

- iron
- iron
insp
- iron
pron
expo
- iron
pyth
ager

Independent

The
fol-
low-
ing
de-
liv-
er-
able
are
re-
lease
in-
de-
pen-
dent

- iron
pyth
ager
buil
- sush
tools
- tenk
- vir-
tu-
albn

Not released

Things to do before releasing

our *standards*, are coherent, and have proper grammar. Combine release notes if necessary (for example, a release note for a feature and another release note to add to that feature may be combined).

The
fol-
low-
ing
de-
liv-
er-
able
do
not
need
to
be
re-
lease

- iron
insp
spec
- iron
spec
- Re-
view
the
un-
re-
lease
re-
lease
note
if
the
proj
uses
then
Mak
sure
they
fol-
low

added since the last release, update the REST API version history (`doc/source/contributor/webapi-version-history.rst`) to indicate that they were part of the new release.

is a named release) into `ironic/common/release_mappings.py`:

• For iron re-lease only not iron insp re-lease if any new API mi-crov sion have been

• To sup-port rolli up-grad add this new re-lease ver-sion (and re-lease nam if it

— in REL. mak a copy of

the new semver release version.

should be the same as that of the latest semver one (that you just added above).

the
mas
en-
try,
and
re-
nam
the
first
mas
en-
try
to

—
If
this
is
a
nam
re-
lease
add
a
REL
en-
try
for
the
nam
re-
lease
Its
valu

It
is
im-
por-
tant
to
do
this
be-
fore
a
sta-
ble/
bran
is

`switch is made` to use the latest release from stable as the old release). Otherwise, once it is made, CI (the grenade job that tests new-release -> master) will fail.

cluding the related documentation.

mad
(or
if
the
gren

•
Che
for
any
oper
patc
that
are
clos
to
be
merg
or
re-
leas
crit-
i-
cal.

This
usu-
ally
in-
clud
im-
por-
tant
bug
fixes
and/
fea-
ture
that
wed
like
to
re-
leas
in-

How to propose a release

itself is almost a 100% automated process, accomplished by following the next steps:

automation resides.

The
step
that
lead
to
a
re-
lease
pro-
posa
are
main
man
ual,
whil
prop
ing
the
re-
lease

- Clon
the
oper
stack
repo
i-
tory.
This
is
whe
de-
liv-
er-
able
are
track
and
all
the

Un-
der
the
del.
di-

subproject) grouped by release cycles.

(official) cycles (e.g. ironic-python-agent-builder).

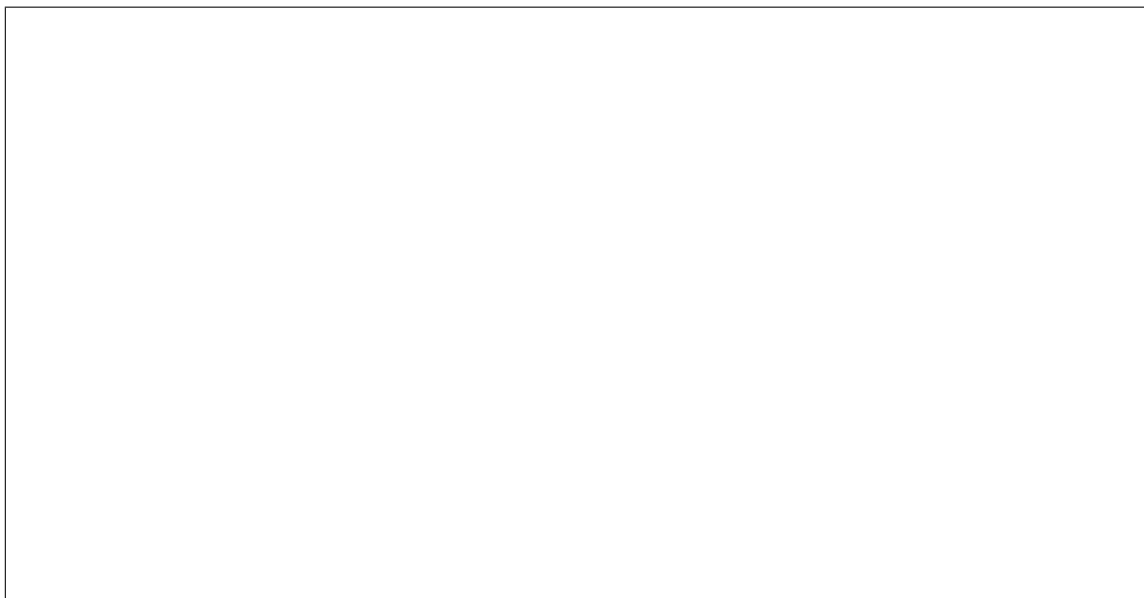
rec-
tory
you
can
see
yam
files
for
each
de-
liv-
er-
able
(i.e.

-
The
_in
di-
rec-
tory
con-
tains
yam
files
for
de-
liv-
er-
able
that
are
not
bound
to

•
To
check
the
char
were
about
to
re-
lease
we
can
use
the

tox
en-
vi-
ron-
men-
lis

with this syntax:



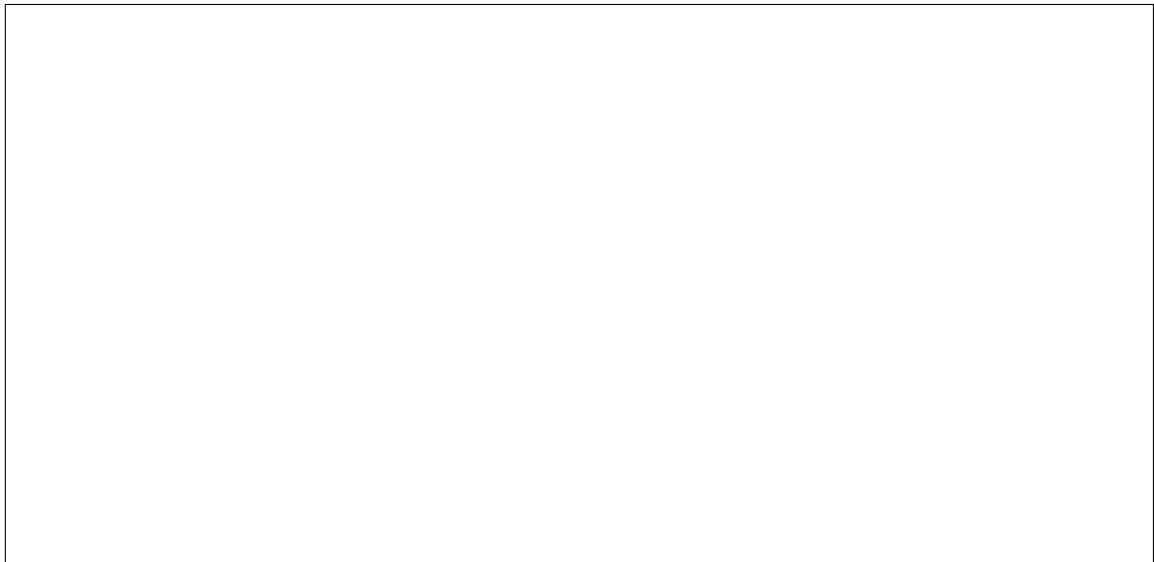
The
ser-
ar-
gu-
men-
is
a
re-
lease
se-
ries
(i.e.
mas-
ter
or
train
not
sta-
ble/
or

stable/train).

For
ex-
am-
ple,
as-

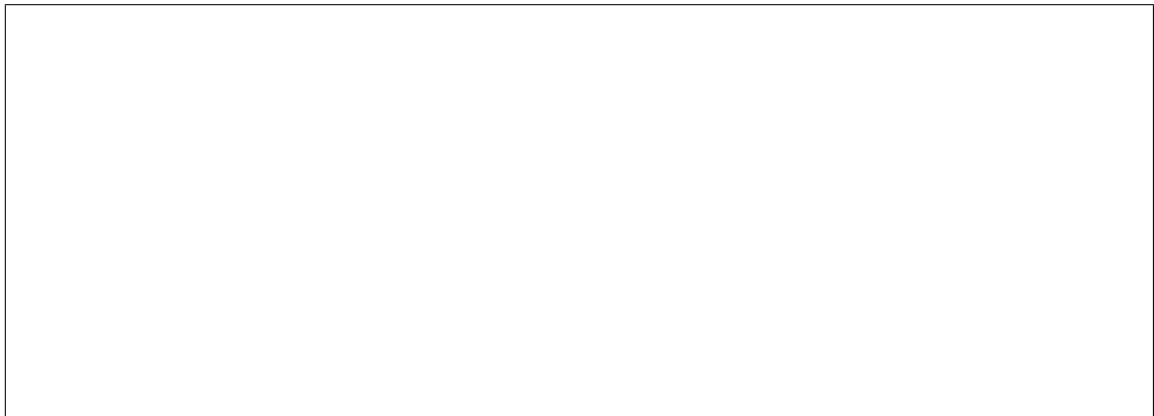
sum
ing
were
in
the
main
di-
rec-
tory
of
the
re-
lease
repo
i-

tory, to check the changes in the ussuri series for ironic-python-agent type:



- To
up-
date
the
de-
liv-
er-
able
file
for
the
new
re-
lease
we
use
a

cess in the form of a tox environment called `new-release`.



need to decide on whether the next version will be major, minor (feature) or patch (bugfix).

scrip
pro-

To
get
fa-
mil-
iar
with
it
and
see
all
the
op-
tion
type

Now
base
on
the
list
of
char
we
foun
in
the
prec
dent
step
and
the
re-
leas
note
we

for the current development branch (master) that takes the code name of the future stable release, for example if the future stable release code name is wallaby, we need to use wallaby as `series`.

projects are not branched this way though.

Note
that
in
this
case
ser
is
a
code
nam
(trai
us-
suri)
not
a
bran
That
is
also
valid

The
--s
arg
is
used
only
for
bran
ing
in
the
end
of
a
cy-
cle,
in-
de-
pen-
dent

The
--i
op-
tion
is
used
to

ing the new release model for ironic projects.

mit the change, and propose it for review.

cre-
ate
an
in-
ter-
me-
di-
ate
bug-
fix
bran
fol-
low-

To
pro-
pose
the
re-
lease
use
the
scrip
to
up-
date
the
de-
liv-
er-
able
file,
then
com

For
ex-
am-
ple,
to
pro-
pose
a
mi-
nor
re-
lease
for
iron
in

rent development branch), considering that the code name of the future stable release is wallaby, use:



erable, the new version and the branch, if applicable.

the
mas
ter
bran
(cur

Re-
men
ber
to
use
a
mea
ing-
ful
topic
usu-
ally
us-
ing
the
nam
of
the
de-
liv-

A
goo
com
mit
mes
sage
ti-
tle

ironic 1.2.3 for ussuri

mitting them for review.

shou
also
in-
clud
the
sam
for
ex-
am-
ple
Re-
leas

- As
an
op-
tion
step
we
can
use
tox
-e
lis
to
doubl
chec
the
char
be-
fore
sub-

Also
tox
-e
val.
(it
migl
take
a
whil
to
run
base
on
the
num
ber

some sanity-checks, but since everything is scripted, there shouldn't be any issue.

doubts or if any errors should arise, you can reach to them in the IRC channel `#openstack-release`; all release liaisons should be present there.

have to approve it before it can get approved by the release team. Then, it will be processed automatically by zuul.

of
char
does
som

All
the
scrip
are
de-
sign
and
main
tain
by
the
re-
leas
team
in
case
of
ques
tion
or

- Af-
ter
the
char
is
up
for
re-
view
the
PTL
or
a
re-
leas
li-
ai-
son
will

Things to do after releasing

When a release is done that results in a stable branch

need to be made.

includes:

When a release is done that results in a stable branch for the project several changes

The release automation will push a number of changes that need to be approved. This

- In the new

sta-
ble
bran

—
a
char
to
poin
.
git:
at
the
bran

—
a
char
to
up-
date
the
up-
per
con-
strai
file
used
by
tox

•
In
the
mas
ter
bran

—
up-
dat-
ing
the
re-
lease
note
RST
to
in-
clud
the
new
bran

ically submit a follow-up patch to do that. An example of this patch is [here](#).

The
gen-
er-
ated
RST
does
not
in-
clud
the
ver-
sion
rang
in
the
ti-
tle,
so
we
typ-

—
up-
date
the
*tem-
plate*
in
.zuul
or
zuul

The
up-
date
is
nec-
es-
sary
to
use
the
job
for
the
next
re-
lease
oper
pyth
<*ne*

An example of this patch is [here](#).

example of this patch is [here](#).

jobs

We need to submit patches for characters in the stable branch to:

- update the iron-devs-tack plugin to point at the branch tarball for IPA. An

- update links in the documentation (irc doc,

versions of any openstack projects (that branch) documents. As of Pike release, the only outlier is [diskimage-builder](#).

unsupported API tempest tests are skipped on stable branches. E.g. [patch 495319](#).

sou
)
to
poin
to
the
bran

- set
ap-
pro-
pri-
ate
de-
fault
for
TEMI
and
TEMI
in
dev.
lib,
iron
to
mak
sure
that

We
need
to
sub-
mit
patc
for
char
on
mas
ter
to:

- cre-
ate
an
emp
com
mit
with

sion. See [example](#) and [pbr documentation](#) for details.

these changes. Note that we need to wait until *after* the switch in grenade is made to test the latest release (N) with master (e.g. [for stable/queens](#)). Doing these changes sooner after the ironic release and before the switch when grenade is testing the prior release (N-1) with master, will cause the tests to fail. (You may want to ask/remind infra/qa team, as to when they will do this switch.)

a
Sem
tag
to
bum
the
gen-
er-
ated
mi-
nor
ver-

- to
sup-
port
rolli
up-
grad
sinc
the
re-
lease
was
a
nam
re-
lease
we
need
to
mak

-
In
iron
com
rel.
py.
dele
any
en-
tries
from
REL
as-

named release. Since we support upgrades between adjacent named releases, the master branch will only support upgrades from the most recent named release to master.

sponding code from ironic. (These migration scripts are used to migrate from an old release to this latest release; they shouldn't be needed after that.)

so-
ci-
ated
with
the
old-
est

—
re-
mov
any
DB
mi-
gra-
tion
scrip
from
iron
cmd
dbs:
ONL
and
re-
mov
the
cor-
re-

—
re-
mov
any
mod
class
nam
from
iron
cmd
dbs:
NEW

Ironic Tempest plugin

master branch. [Example for Queens](#).

Bifrost

[Example for Victoria](#). The upper constraints file referenced in `scripts/install-deps.sh` needs to be updated to the new release.

As
iron
tem
plug
is
bran
less,
we
need
to
sub-
mit
a
patc
addi
sta-
ble
jobs
to
its

Bifr
need
to
be
up-
date
to
in-
stall
de-
pen-
den-
cies
us-
ing
the
sta-
ble
bran
Ex-

For all releases

For
all
re-
lease
when
or
not
it
re-
sults
in
a
sta-
ble
bran

- up-
date
the
spec
repo
to
marl
any
spec
com
plete
in
the
re-
lease
as
im-
ple-
men

- re-
mov
any
-
2s
on
patc
that
were
bloc
un-
til

af-
ter
the
re-
lease

Ironic Governance Structure

The
ironic
project
manages
a
number
of
repositories
i-
to-
ries
that
con-
tribute
to
our
mission

The full list of repositories that ironic manages is available in the [governance site](#).

What belongs in ironic governance?

For
a
repository
i-
tory
to
be
part
of
the
Ironic
project

- It
must
com-
ply

brary that implements a standard to manage hardware from multiple vendors (such as IPMI or redfish) is okay.

with
the
TCs
rules
for
a
new
proj

- It must not be intended for use with only a single vendor's hardware. A li-

- It must align with Ironics' mission statement.

Lack of contributor diversity

itory where only a single company is contributing is okay, with the hope that other companies will contribute after joining the ironic project.

Proposing a new project to ironic governance

is
a
chic
egg
prob
lem,
and
as
such
a
repo

Repe
i-
to-
ries
that
are
no
long
main
taine
shou
be
prun
from
gov-
er-
nanc
reg-
u-
larly

Brin
the
pro-
posa
to
the
iron
weel
mee
ing
to
dis-
cuss
with

the
team

8.1. Dri

Iron
ics
com
mu-
nity
in-
clud
man
hard
ware
ven-
dors
who
con-
tribu
drive
that
en-
able
more

advanced functionality when IroniC is used in conjunction with that hardware. To do this, the IroniC developer community is committed to standardizing on a [Python Driver API](#) that meets the common needs of all hardware vendors, and evolving this API without breaking backwards compatibility. However, it is sometimes necessary for driver authors to implement functionality - and expose it through the REST API - that can not be done through any existing API.

To
fa-
cil-
i-
tate
that,
we
also
pro-
vide
the
mea
for
API
calls
to
be
pass
thro

and directly to the driver. Some guidelines on how to implement this are provided below. Driver authors are strongly encouraged to talk with the developer community about any implementation using this functionality.

Pluggable Drivers

drivers, and operators to use third-party drivers or write their own. A driver is built at runtime from a *hardware type* and *hardware interfaces*. See *Enabling drivers and hardware types* for a detailed explanation of these concepts.

from the setuptools entrypoints `ironic.hardware.types` and `ironic.hardware.interfaces.<INTERFACE>` where `<INTERFACE>` is an interface type (for example, `deploy`). Only hardware types listed in the configuration option `enabled_hardware_types` and interfaces listed in configuration options `enabled_<INTERFACE>_interfaces` are loaded. A complete list of hardware types available on the system may be found by enumerating this entrypoint by running the following python script:

```
from setuptools import entry_points
from ironic.common import utils

def main():
    """Enumerate hardware types and interfaces"""
    hw_types = entry_points(group='ironic.hardware.types')
    hw_interfaces = entry_points(group='ironic.hardware.interfaces')

    for hw_type in hw_types:
        print(hw_type.name)

    for hw_interface in hw_interfaces:
        print(hw_interface.name)

if __name__ == '__main__':
    main()
```

```
→ #
→ /
→ u
→ b
→ e
→ p
imp
→ pl
→ a
→ pl
prin
→ [
→ n
→ f
→ p
→ i
→ pl
→ i
→ e
→ p
→ "
→ h
→ t
→ "
→ i
→ n
→ p
→ n
→ s
→ "
→ "
```

A list of drivers enabled in a running Iron

the following command against that API end point:

Writing a hardware type

setuptools entry point `ironic.hardware.types`. Most of the real world hardware types inherit `ironic.drivers.generic.GenericHardware` instead. This helper class provides useful implementations for interfaces that are usually the same for all hardware types, such as `deploy`.

implementation is provided by the `GenericHardware` base class.

are provided by the `GenericHardware` base class.

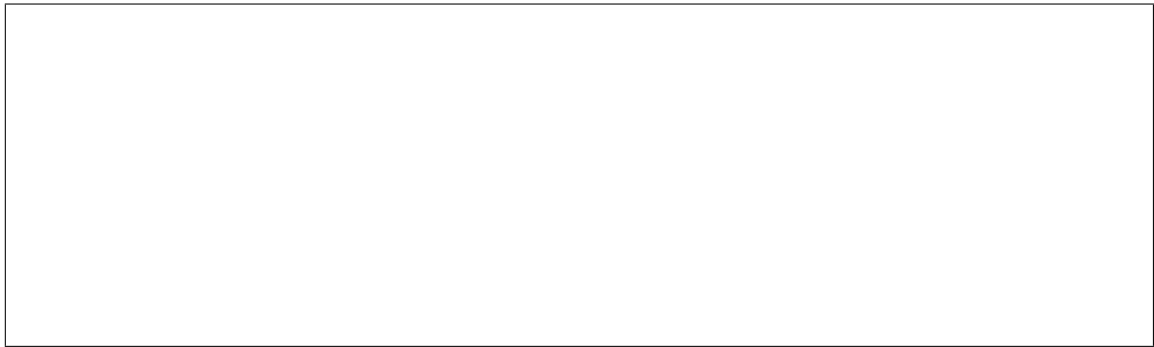
- *boot*
that
spec
i-
fies
how
to
boot
rame
and
in-
stan
on
the
hard
ware
A
gene
pxe

- *de-
ploy*
that
or-
ches
trate
the
de-
ploy
men
A
few
com
mon
im-
ple-
men
ta-
tions

As
of
the
Rock
re-
lease
a
de-

ploy
in-
ter-
face
shou
dec-
o-
rate
its
de-
ploy
meth

to indicate that it is a deploy step. Conventionally, the deploy method uses a priority of 100.



Note
Mos
of
the
hard
ware
type
shou
not
over
ride
this
in-
ter-
face

- *pow*
im-
ple-
men
pow
ac-
tions
for
the

may be used, if supported by the hardware:

`base.PowerInterface` and providing missing methods.

hard
ware
The
com
mon
im-
ple-
men
ta-
tions

-
iron
dri
mod
ipm
IPM

-
iron
dri
mod
red
pow
Red

Oth-
er-
wise
you
need
to
writ
your
own
im-
ple-
men
ta-
tion
by
sub-
class
ing
iron
dri

Note
Pow
ac-

not return until the power action is finished or errors out.

setting a boot device. A few common implementations exist and may be used, if supported by the hardware:

tions
in
Iron
are
bloc
ing
-

meth
ods
of
a
pow
in-
ter-
face
shou

- *man*
age-
men
im-
ple-
men
ad-
di-
tiona
out-
of-
band
man
age-
men
ac-
tion
such
as

- *iro*
dri
mod
ipm
IPM

- *iro*
dri

fake implementation in `ironic.drivers.modules.fake.FakeManagement` instead.

`base.ManagementInterface` and providing missing methods.

mod
red
man
Red

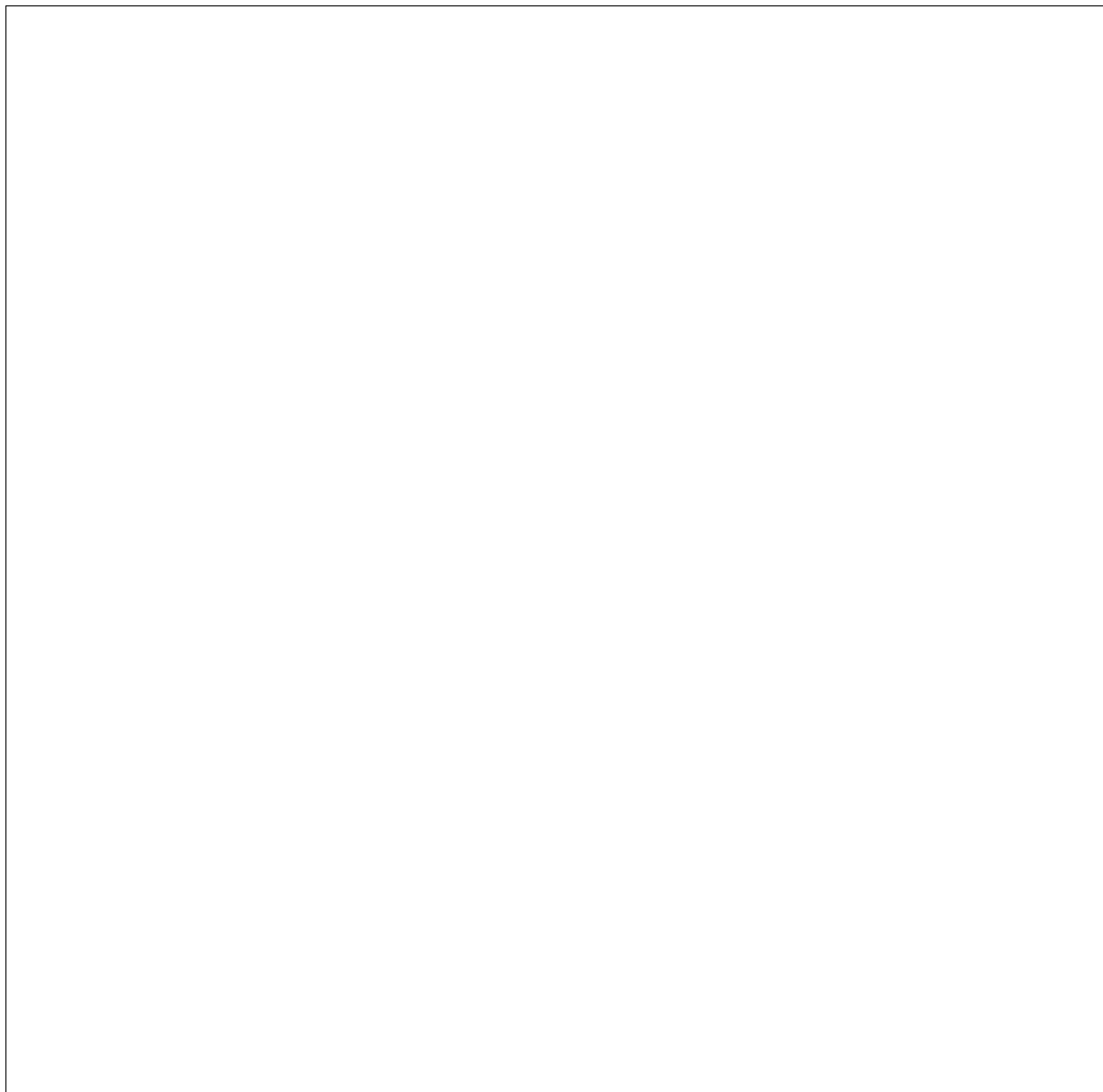
Som
hard
ware
type
such
as
snmp
do
not
sup-
port
out-
of-
band
man
age-
men
They
use
the

Oth-
er-
wise
you
need
to
writ
your
own
im-
ple-
men
ta-
tion
by
sub-
class
ing
iro
dri

Com
bine
the
in-

ter-
face
in
a
hard
ware
type
by
pop-
u-
lat-
ing
the
lists
of
sup-

ported interfaces. These lists are prioritized, with the most preferred implementation first. For example:



cla
→ M
→ G

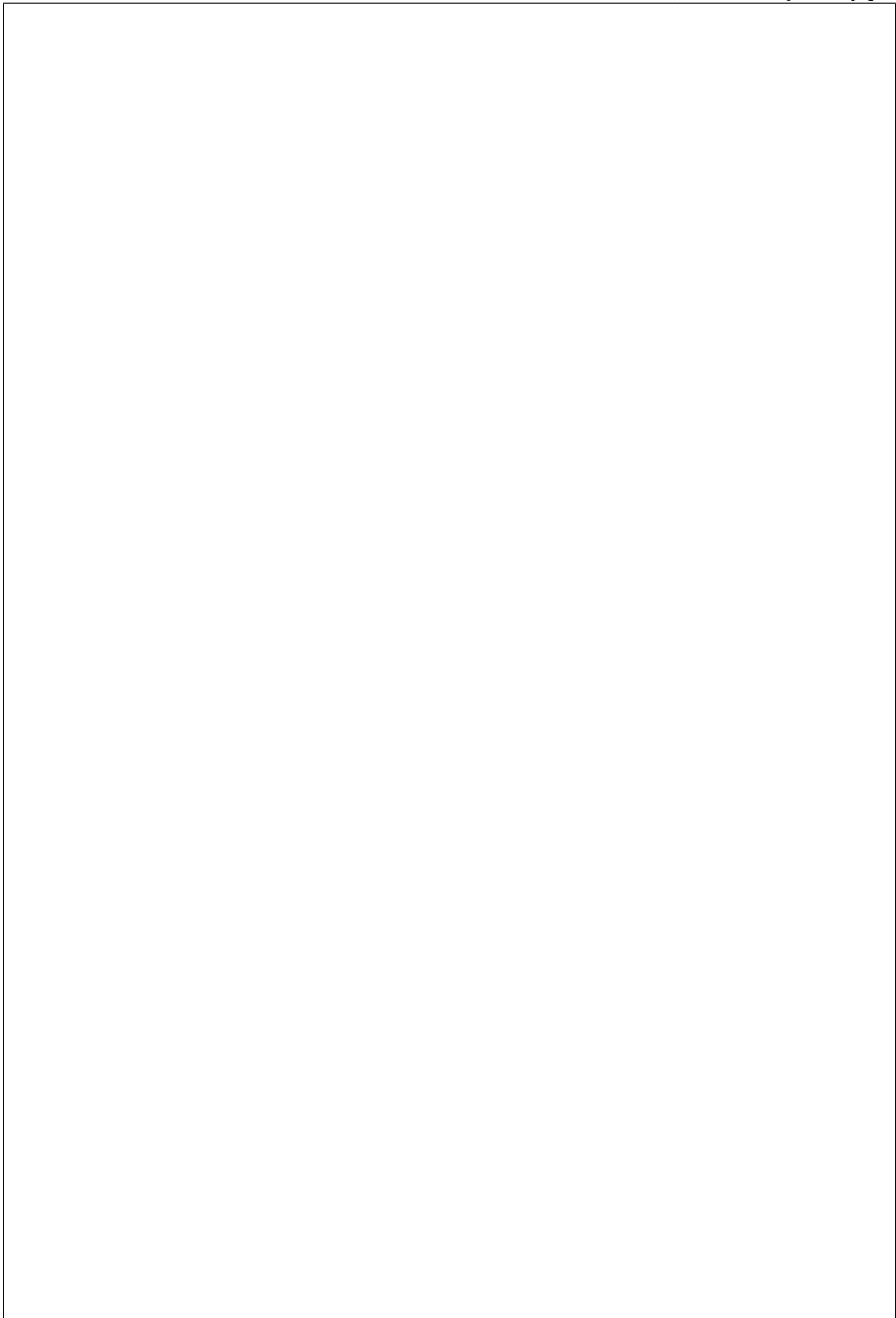
└
→ └
→ └
→ └
→ @

└
→ └
→ └
→ └
→ d
→ S
→ m
→ i

└
→ └
→ └
→ └
→ └
→ └
→ └
→ └
→ └
→
→ "
→ "
→ "
→ O
→ S
→ m
→ i
→
→ "
→ "
→ "

(continues on next page)

(continued from previous page)

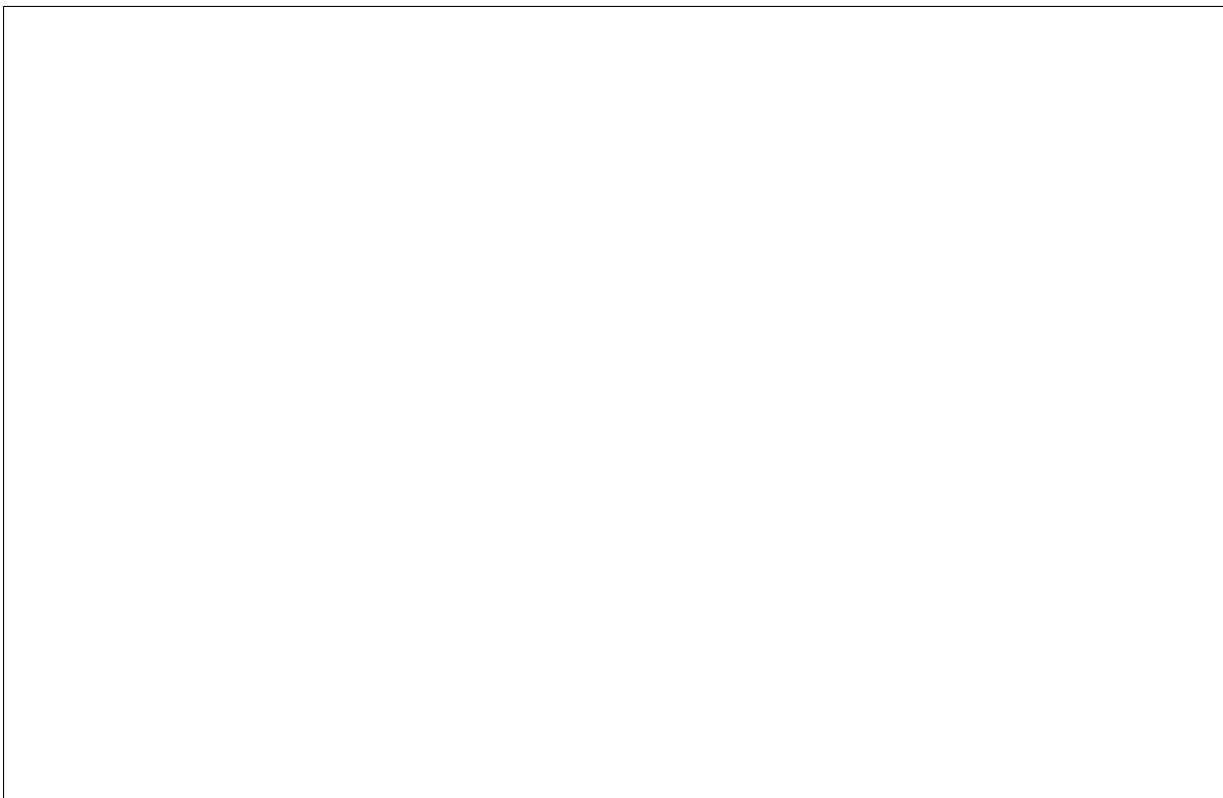


(continues on next page)

(continued from previous page)

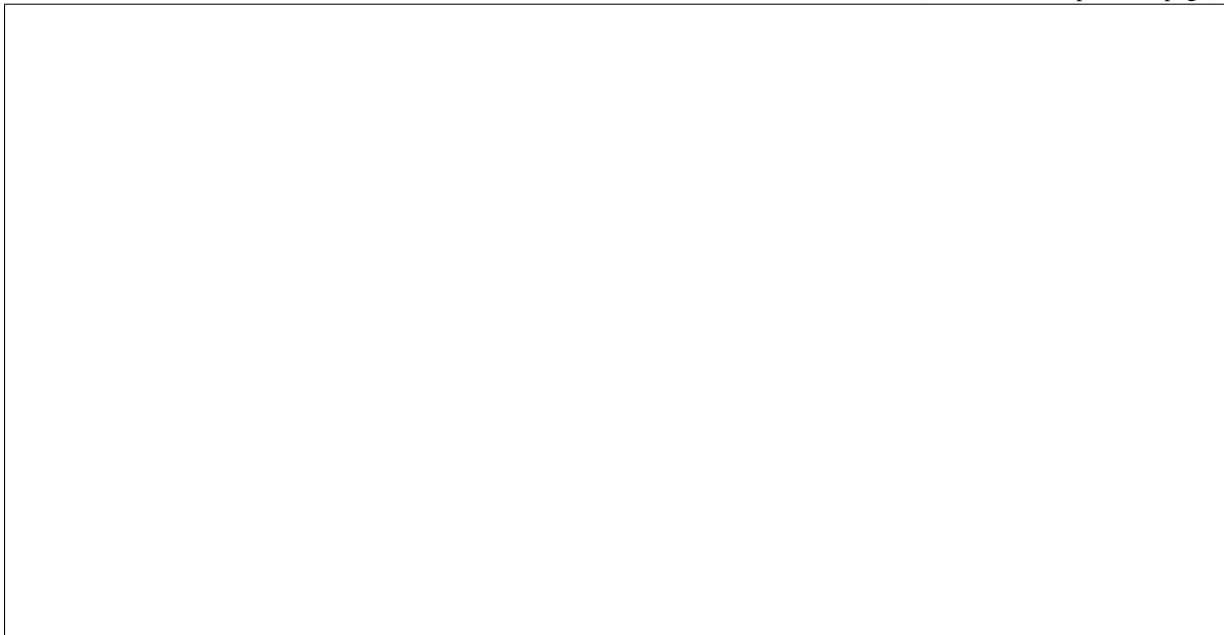
Note: In this example, all interfaces, except for management and power are taken from the GenericHardware base class.

try points for them in the setup.cfg file:



(continues on next page)

(continued from previous page)



Supported Drivers

commit) please consult the *drivers page*.

iron
→ h
→ i
→ m
→
↳
→ ↳
→ ↳
→ ↳
→ m
→ m
→ =
→ i
→ d
→ m
→ m
→ h

For
a
list
of
sup-
port
driv
(tho
that
are
con-
tin-
u-
ousl
teste
on
ev-
ery
up-
strea

Node Vendor Passthru

UUID or Name>/vendor_passthru?method={METHOD} endpoint. Beyond basic checking, Ironic does not introspect the message body and simply passes it through to the relevant driver.

Driv
may
im-
ple-
men
a
pass
API
whic
is
ac-
ces-
si-
ble
via
the
/
v1/
nod
<No

A
meth

- can support one or more HTTP methods (for example, GET, POST)

- is asynchronous or synchronous

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

that the request was fulfilled. The response may include a body.

—
For
asyn
chro
meth
ods,
a
202
(Ac-
cept
HTT
sta-
tus
code
is
re-
turn
to
in-
di-

—
For
syn-
chro
meth
ods,
a
200
(OK
HTT
sta-
tus
code
is
re-
turn
to
in-
di-
cate

•
can
re-
quir
an
ex-
clu-
sive

specify `require_exclusive_lock=False` in the decorator. If an exclusive lock is held on the node, other requests for the node will be delayed and may fail with an HTTP 409 (Conflict) error code.

Ironics standard REST API. There is only a single HTTP endpoint exposed, and the semantics of the message body are determined solely by the driver. Ironic makes no guarantees about backwards compatibility; this is solely up to the discretion of each drivers author.

lock
on
the
node
This
only
oc-
curs
if
the
meth
does

This
end-
point
ex-
pose
a
node
driv
di-
rectl
and
as
such
it
is
ex-
pres
not
part
of

To
get
in-
for-
ma-
tion
about
all
the
meth
ods
avai
able

a particular node, you can issue an HTTP GET request:



name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

via
the
ven-
dor_
end-
poin
for

GET,
↪/
↪v
↪n
↪
↪<
↪U
↪O
↪n
↪n
↪/
↪v
↪p
↪m

The
re-
spon
JSON
body
will
con-
tain
in-
for-
ma-
tion
for
each
meth
such
as
the
meth
ods

Driver Vendor Passthru

`<driver name>/vendor_passthru?method={METHOD}`.

Driv
may
im-
ple-
men
an
API
for
re-
ques
not
re-
latec
to
any
node
at
/
v1/
dri

A
meth

- can support one or more HTTP methods (for example, GET, POST)
- is asynchronous or synchronous

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

that the request was fulfilled. The response may include a body.

Note: Unlike methods in *Node Vendor Passthru*, a request does not lock any resource, so it will not delay other requests and will not fail with an HTTP 409 (Conflict) error code.

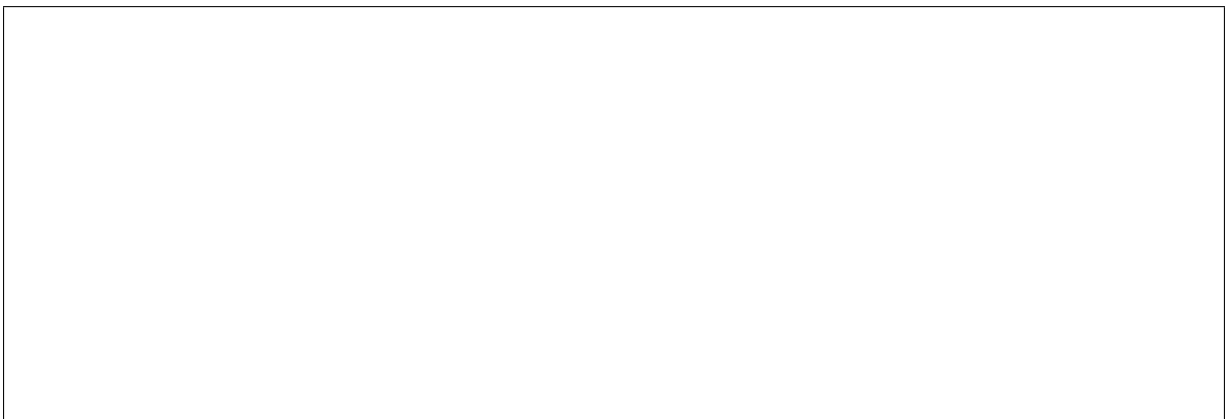
For
asyn
chro
meth
ods,
a
202
(Ac-
cept
HTT
sta-
tus
code
is
re-
turn
to
in-
di-

For
syn-
chro
meth
ods,
a
200
(OK
HTT
sta-
tus
code
is
re-
turn
to
in-
di-
cate

Iron
mak
no
guan
an-

point. That is left up to each drivers author.

you can issue an HTTP GET request:



tees
about
the
se-
man
tics
of
the
mes
sage
BO
sent
to
this
end-

To
get
in-
for-
ma-
tion
about
all
the
meth
ods
avai
able
via
the
driv
ven-
dor_
end-
point

GET,
→/
→v
→d
→
→<
→n
→/
→v
→p
→m

The

name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

Vendor Methods

a driver.

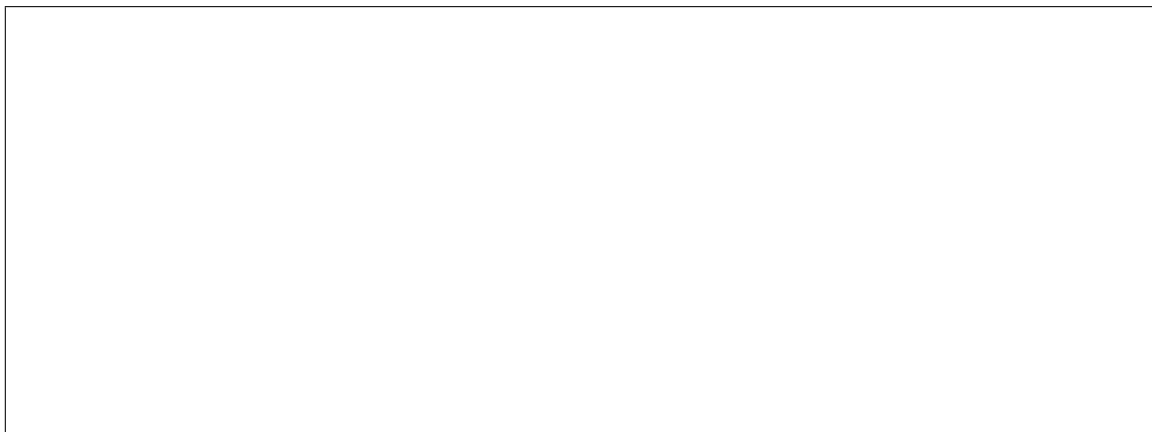
re-
spor
JSON
body
will
con-
tain
in-
for-
ma-
tion
for
each
meth
such
as
the
meth
ods

This
doc-
u-
men
is
a
quic
tu-
to-
rial
on
writ
ing
ven-
dor
spe-
cific
meth
ods
to

The
first
thing
to
note
is

node vendor passthru and a node vendor passthru.

which is not specific to a Node. For example, lets say the driver *ipmi* exposed a method called *authentication_types* that would return what are the authentication types supported. It could be accessed via the Ironic API like:



(continues on next page)

that
the
Iron
API
sup-
port
two
ven-
dor
end-
poin
A
driv
ven-

•
The
Vene
al-
lows
hard
ware
type
to
ex-
pose
a
cus-
tom
top-
leve
func
tion-
al-
ity

(continued from previous page)

terface implementation for a given hardware type. This limitation will be lifted in the future.

node basis. For example the same driver *ipmi* exposing a method called *send_raw* that would send raw bytes to the BMC, the method also receives a parameter called *raw_bytes* which the value would be the bytes to be sent. It could be accessed via the Ironic API like:

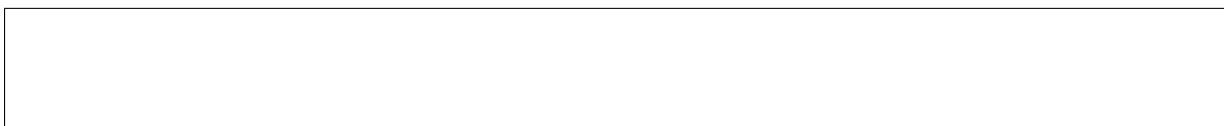
V
T
E
M
A
c
r
o
a
l
t
u
d
p
f
t
d
f
v
i

•
The
node
ven-
dor
pass
al-
lows
drive
to
ex-
pose
cus-
tom
func
tion-
al-
ity
on
per-



Writing Vendor Methods

write a class inheriting from the `VendorInterface` class:



(continues on next page)

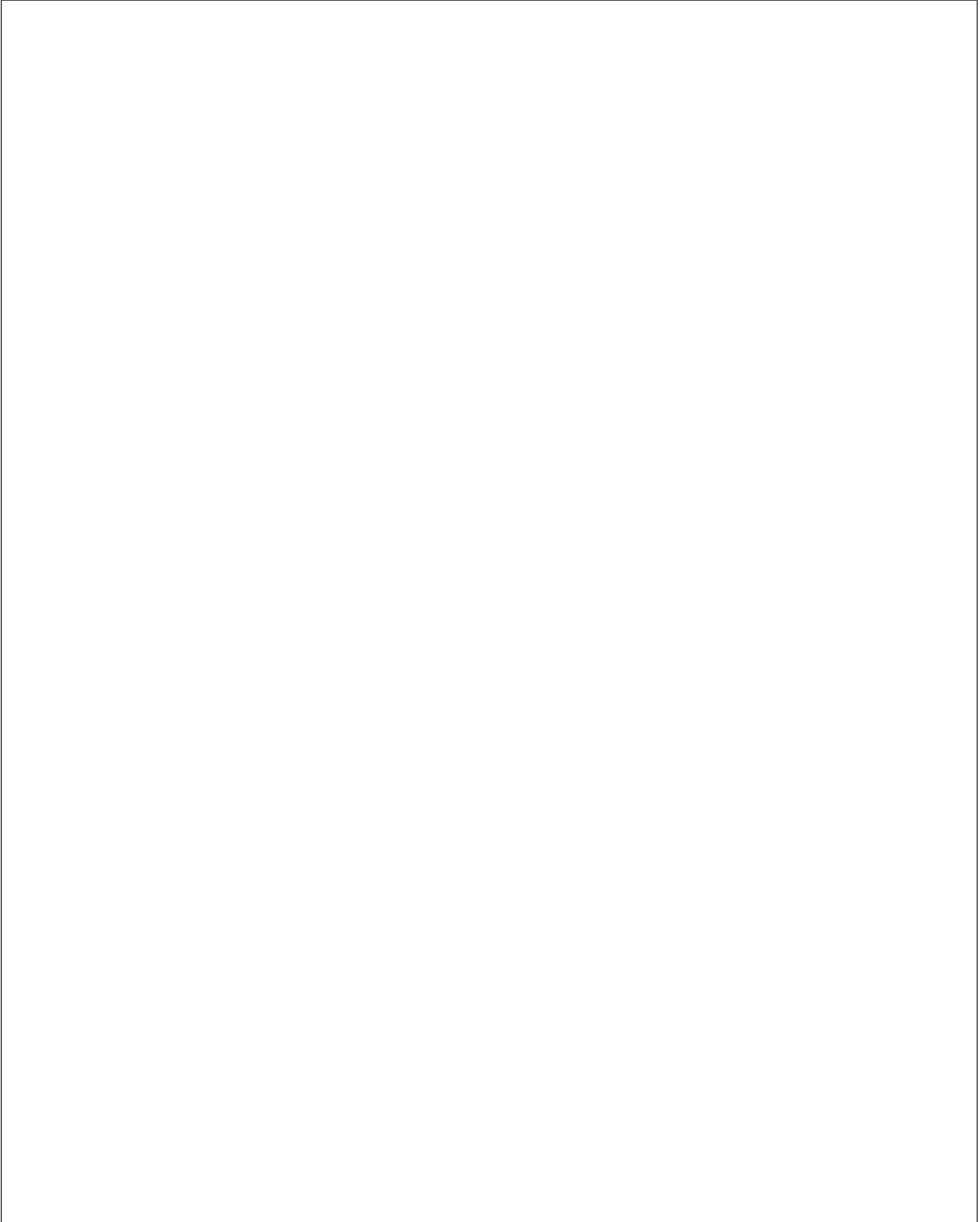
Writ
ing
ing
a
cus-
tom
ven-
dor
meth
in
Iron
shou
be
sim-
ple.
The
first
thing
to
do
is

cla
↔ E.

of <property>:<description> telling in the description whether that property is required or optional so the node can be manageable by that driver. For example, a required property for a *ipmi* driver would be *ipmi_address* which is the IP address or hostname of the node. We are returning an empty dictionary in our example to make it simpler.

passed to the vendor methods. Ironic will not introspect into what is passed to the drivers, its up to the developers writing the vendor method to validate that data.

tion_types which will be exposed on the driver vendor passthru endpoint; And the *send_raw* method that will be exposed on the node vendor passthru endpoint:



(continues on next page)

ods,
the
au-
then
ti-
ca-

cla:

→ [E](#)

→

→

→

→

→ **d**

→ **g**

→ **p**

→

→

→

→

→

→

→

→

→ **r**

→

→ {

→

→

→

→

→ **d**

→ **v**

→

→ t

→

→ m

→

→ *

→

→

→

→

→

→

→

→

→

→ **i**

→ m

→ =

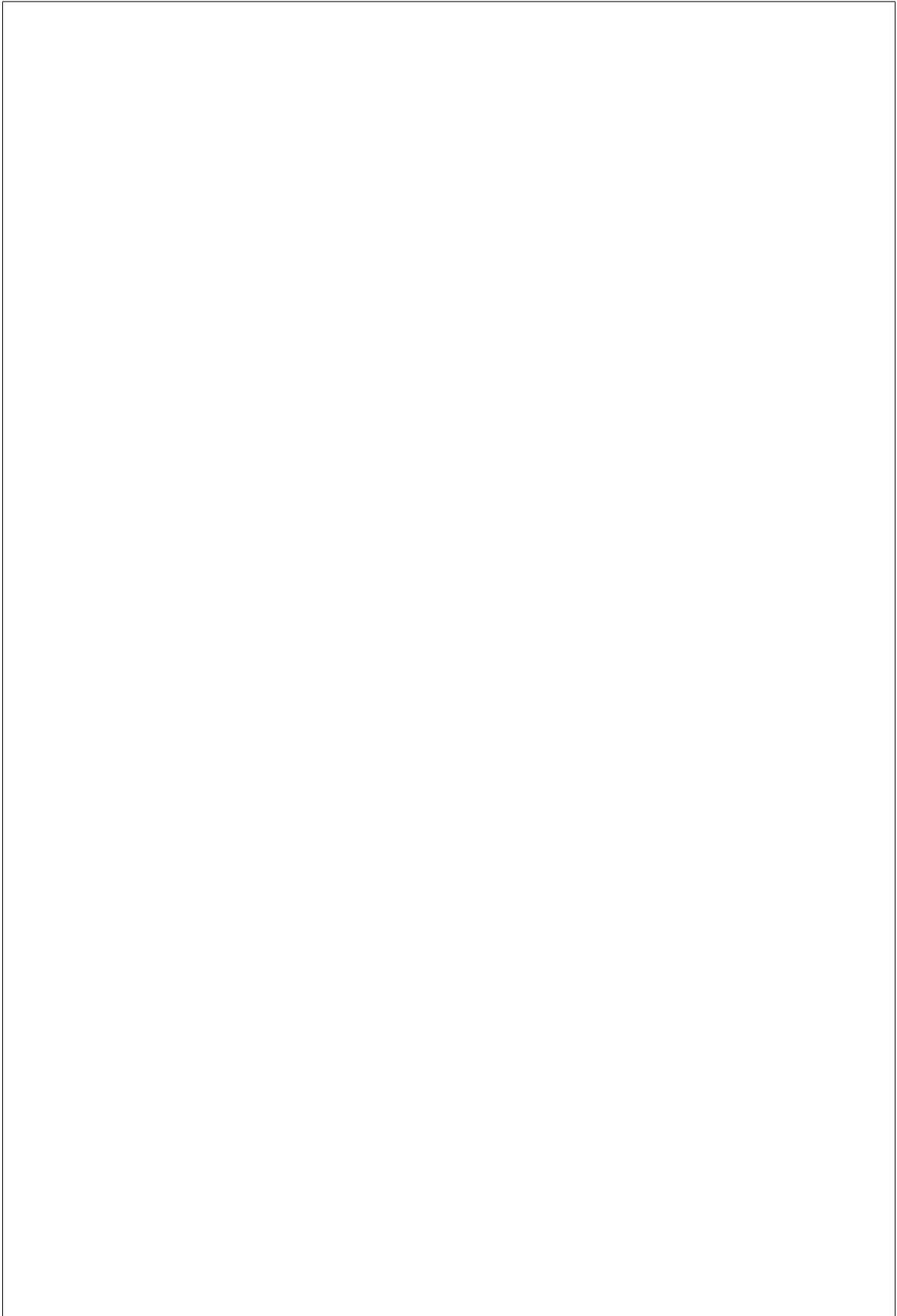
→

→

→

→

(continued from previous page)



(continues on next page)

(continued from previous page)

```
↪ "MD2"]}]
```

(continues on next page)

↪
↪
↪
↪
↪ d
↪ a
↪ t
↪
↪ C
↪
↪ *
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪ r
↪
↪
↪ {
↪ "
↪ "
↪ [
↪ "
↪ "
↪ "
↪
↪
↪ "
↪ "
↪
↪
↪
↪ "
↪ "
↪
↪
↪
↪
↪
↪
↪
↪
↪ @
↪ p
↪ '
↪ '
↪
↪
↪
↪
↪
↪ d
↪ S
↪ r
↪
↪
↪ t
↪
↪
↪ *

(continued from previous page)



r
b
k
g
'
b
'
.
.br/>.br/>.br/>.br/>.br/>.br/>.br/.

That
it!
Writ
ing
a
node
or
drive
ven-
dor
pass
meth
is
prett
muc
the
sam
the
only
dif-

is how you decorate the methods and the first parameter of the method (ignoring self). A method decorated with the `@passthru` decorator should expect a Task object as first parameter and a method decorated with the `@driver_passthru` decorator should expect a Context object as first parameter.

what HTTP method that function was invoked with, a `http_method` parameter will be present in the `kwargs`. Supported HTTP methods are `POST`, `PUT`, `GET` and `PATCH`.

fer-
ence

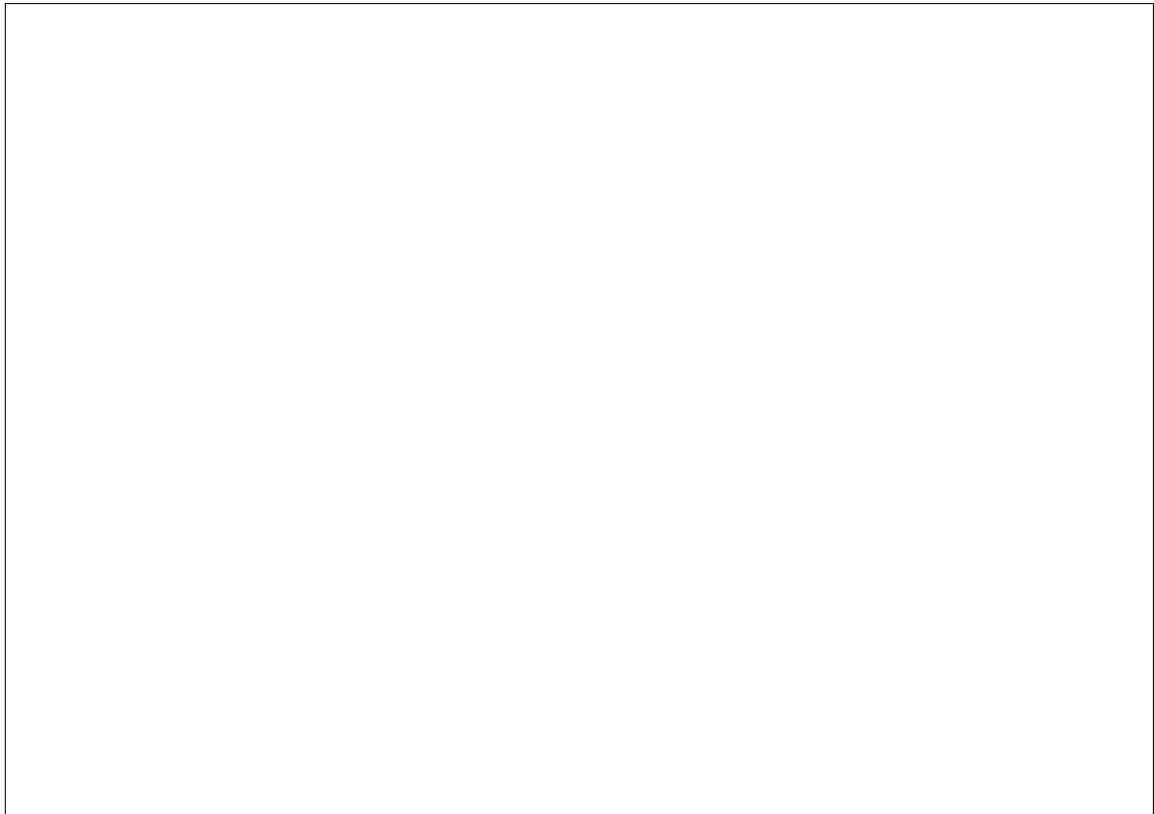
Both
dec-
o-
ra-
tors
ac-
cept
thes
pa-
ram-
e-
ters:

- [http](#)
A
list
of
wha
the
HTT
meth
ods
sup-
port
by
that
ven-
dor
func
tion.
To
know

- [meth](#)
By
de-
fault
the
meth
nam
is
the
nam

of
the
pyth
func
tion.
if
you
wan
to

use a different name this parameter is where this name can be set. For example:



- de-
scrip
tion:
A
strin
con-
tain-
ing
a
nice
de-
scrip
tion
about
wha
that

posed to do. Defaults to (empty string).

faults to True (Asynchronously).

meth
is
sup-

- asyn
A
bool
valu
to
de-
ter-
mine
when
this
meth
shou
run
asyn
chro
or
syn-
chro
De-

Note
This
pa-
ram-
e-
ter
was
pre-
vi-
ously
called
asyn

The
node
ven-
dor
pass
dec-
o-
ra-
tor
(@p
also

ter:

lock on a node between `validate()` and the beginning of method execution. For synchronous methods, the lock on the node would also be kept for the duration of method execution. Defaults to `True`.



ac-
cept
the
fol-
low-
ing
pa-
ram-
e-

- re-
quir
A
bool
valu
de-
ter-
min-
ing
whe
this
meth
shou
re-
quir
an
ex-
clu-
sive

War
Plea
avoi
hav-
ing
a
syn-
chro
meth
for
slow
runn
op-
er-
a-
tion

does talk to a BMC; BMCs are flaky and very easy to break.

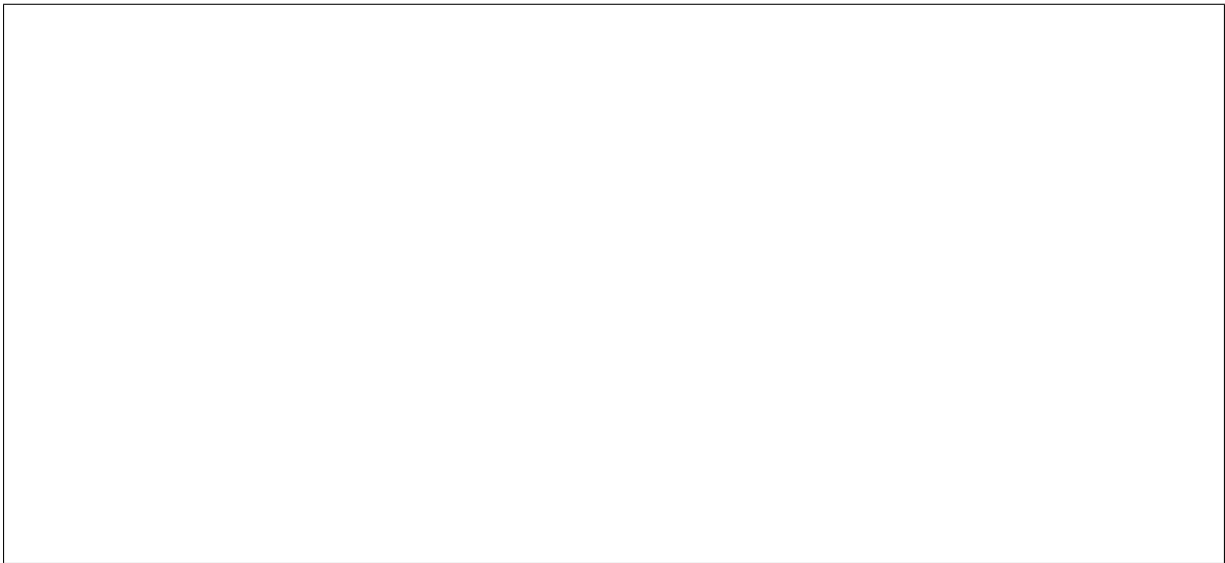
starvation of the thread pool, resulting in a denial of service.

or
if
the
meth

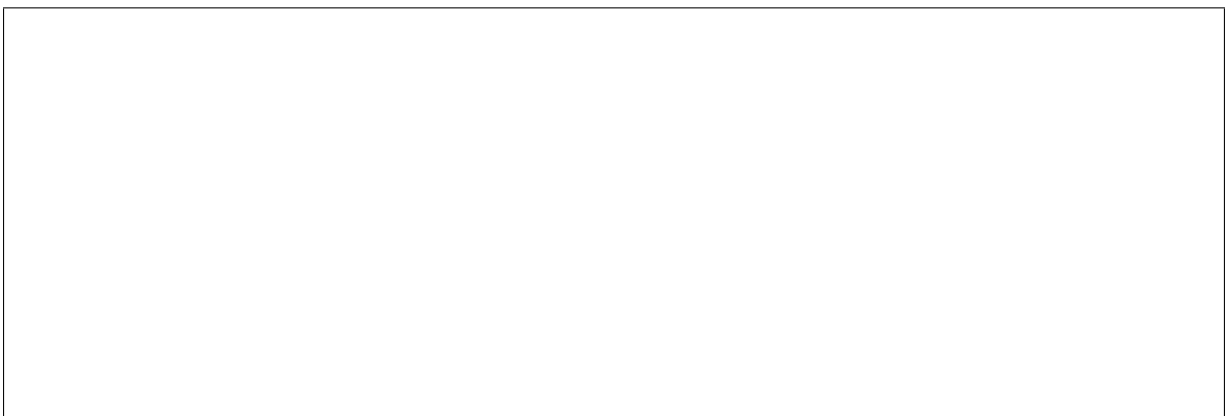
War
Each
asyn
chro
re-
ques
con-
sum
a
worl
threa
in
the
iron
pro-
cess
This
can
lead
to

Give
the
new
ven-
dor
in-
ter-
face
im-
ple-
men
ta-
tion
a
hum
frier
nam
and
cre-
ate

an entry point for it in the `setup.cfg`:



ware types, for example:



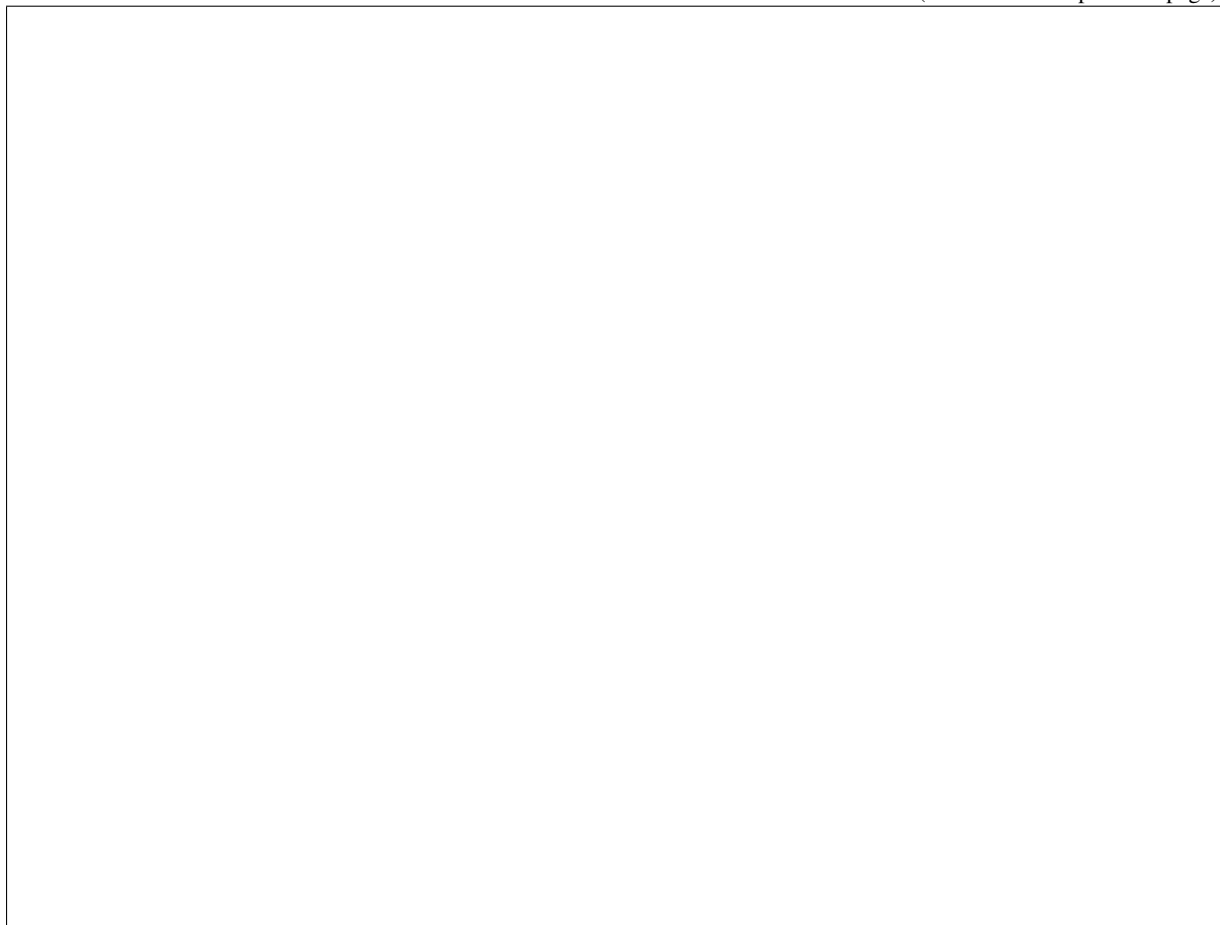
(continues on next page)

iron
→ h
→ i
→
→ =
↳
→↳
→↳
→↳
→ e
→ =
→ i
→ d
→ m
→ e

Fi-
nally
add
it
to
the
list
of
sup-
port
ven-
dor
in-
ter-
face
for
rel-
e-
vant
hard

cla
→ E
→ G
↳
→↳
→↳
→↳
→ .
→ .
→ .
→

(continued from previous page)



Backwards Compatibility

However, for your users' sakes, we highly recommend that you do so.

↳
↳↳
↳↳
↳↳
↳@
↳
↳↳
↳↳
↳↳
↳d
↳S
↳v
↳i
↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳r
↳[
↳E

There
is
no
re-
quir
men
that
char
to
a
ven-
dor
meth
be
back
war
com
pat-
i-
ble.

same HTTP code is being returned to the user.

cates this.

If
you
are
chan
ing
the
ex-
cep-
tions
be-
ing
raise
you
migl
wan
to
en-
sure
that
the

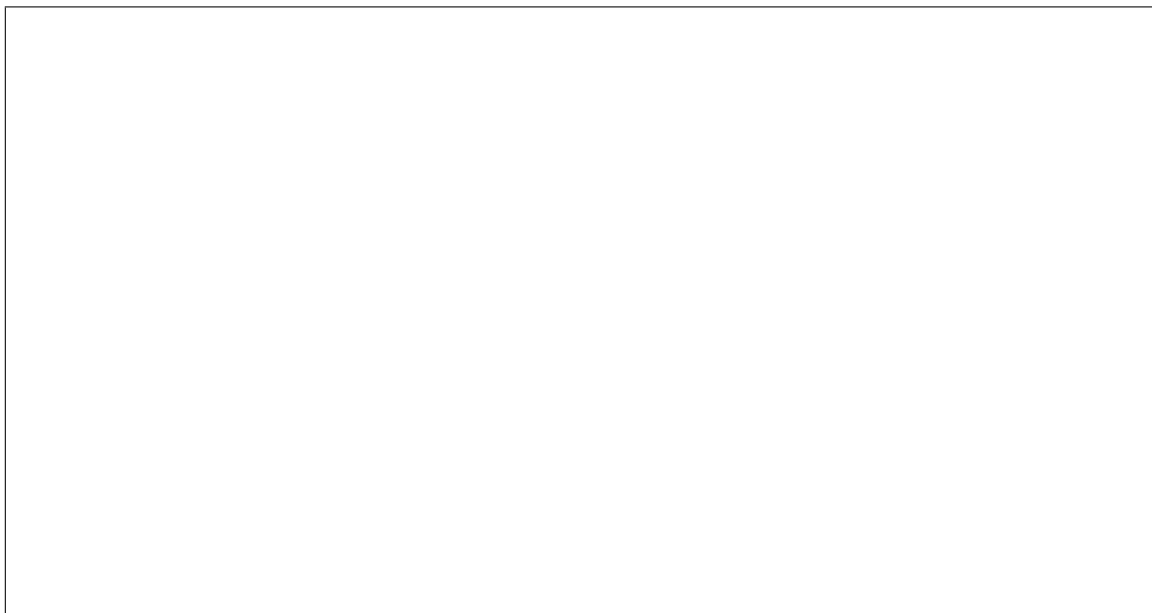
For
non-
back
com
pat-
i-
bil-
ity,
plea
mak
sure
you
add
a
re-
lease
note
that
in-
di-

in-
ter-
face
have
The
hard
ware
in-
ter-
face

that supports BIOS settings should also implement the following three methods:

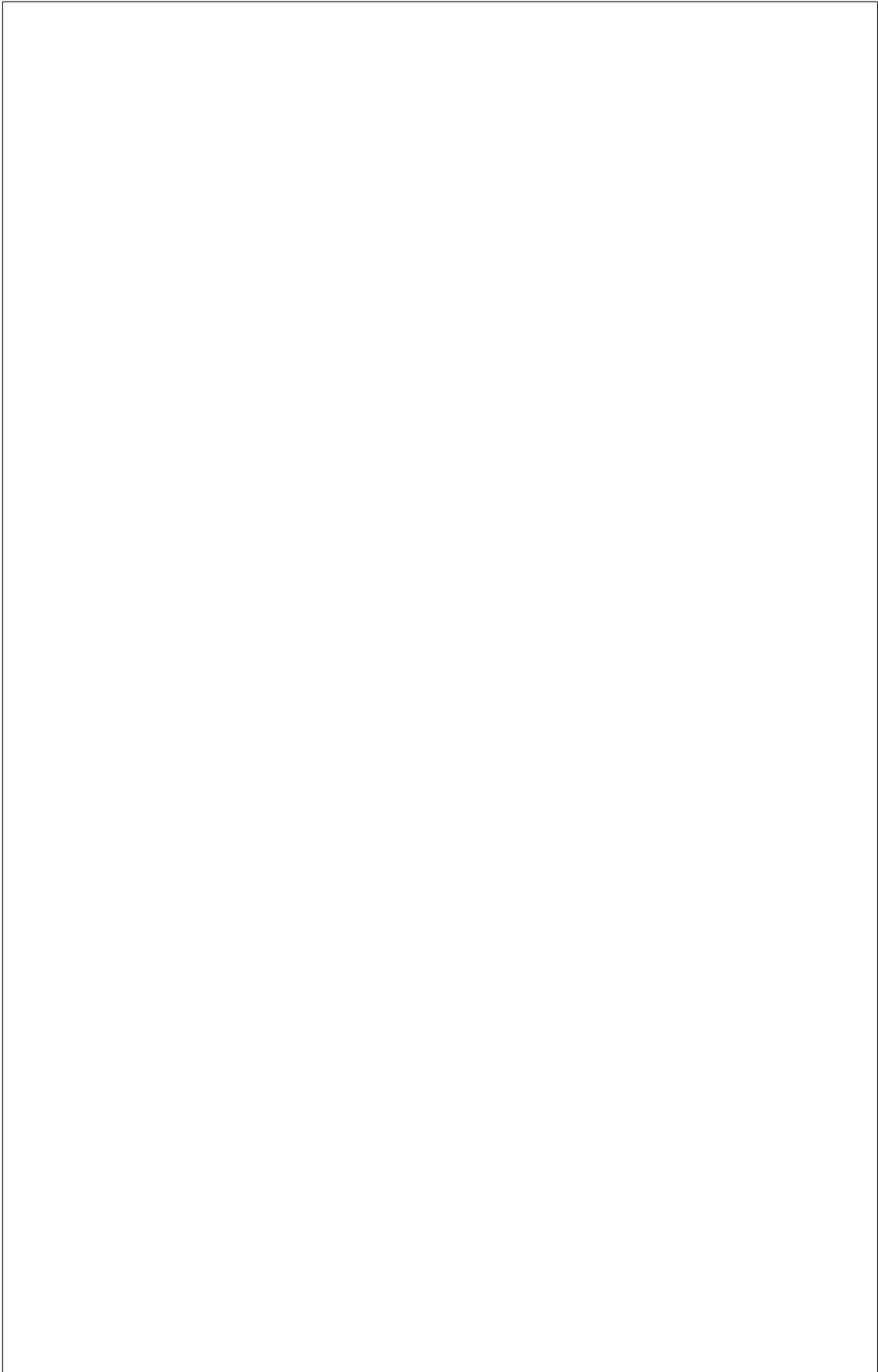
- Im-
ple-
men
a
meth
nam
cach
This
meth
stor
BIO
set-
ting
to
the
bio
ta-
ble
dur-

ing cleaning operations and updates the `bios_settings` table when `apply_configuration` or `factory_reset` are successfully called.



(continues on next page)

(continued from previous page)



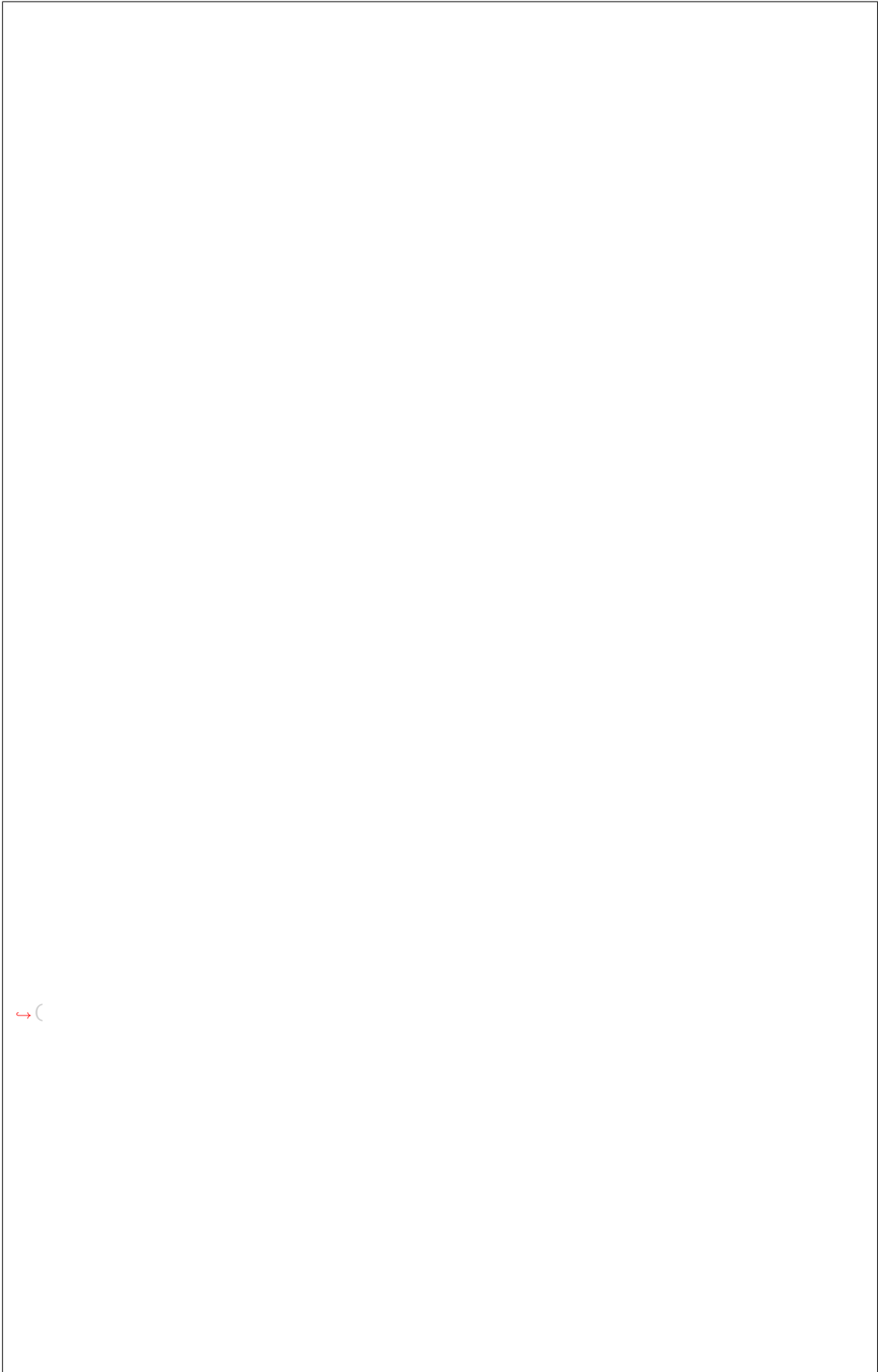
(continues on next page)

(continued from previous page)

```
↪import driver library"))
```

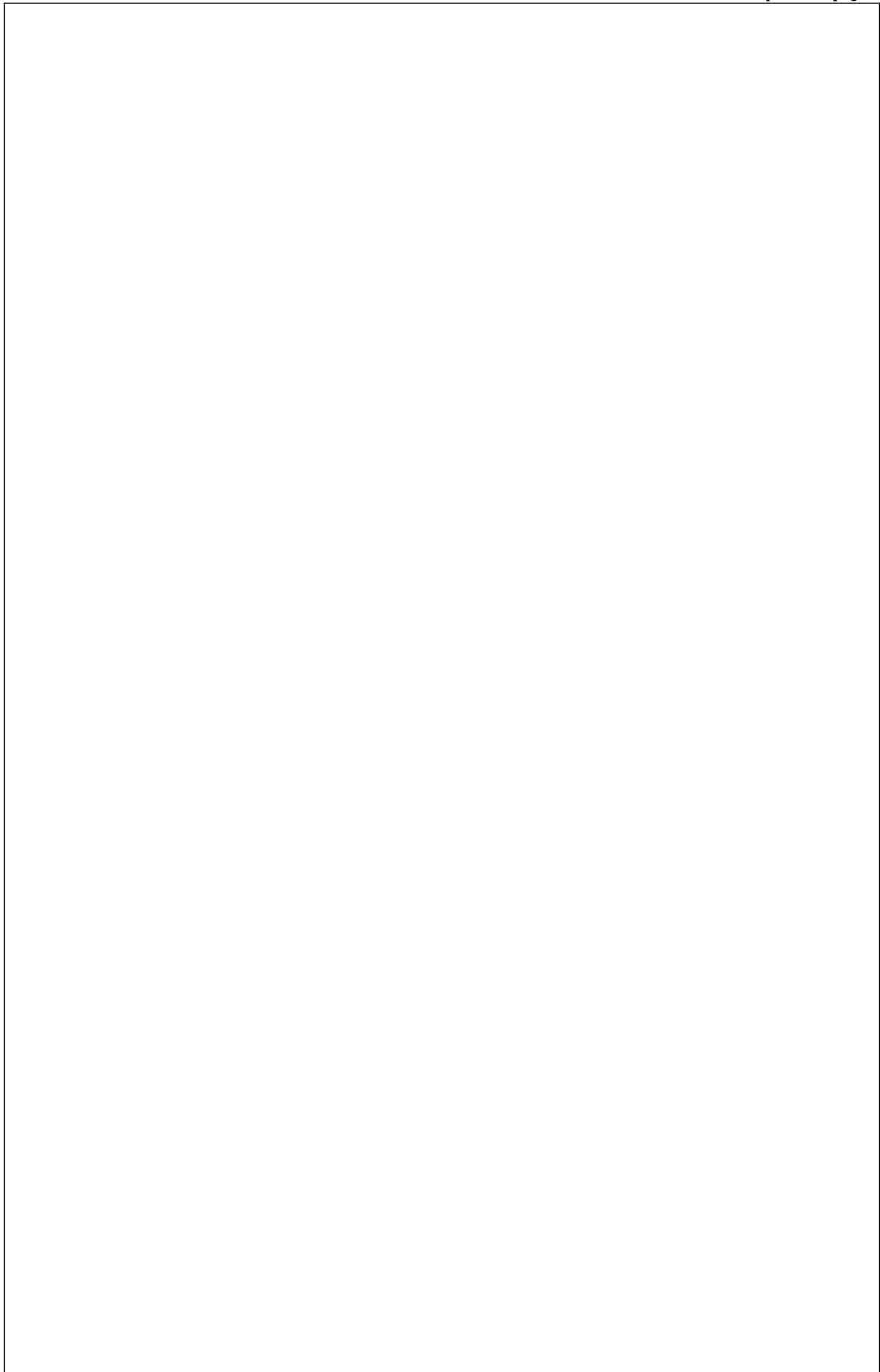
(continues on next page)

(continued from previous page)



(continues on next page)

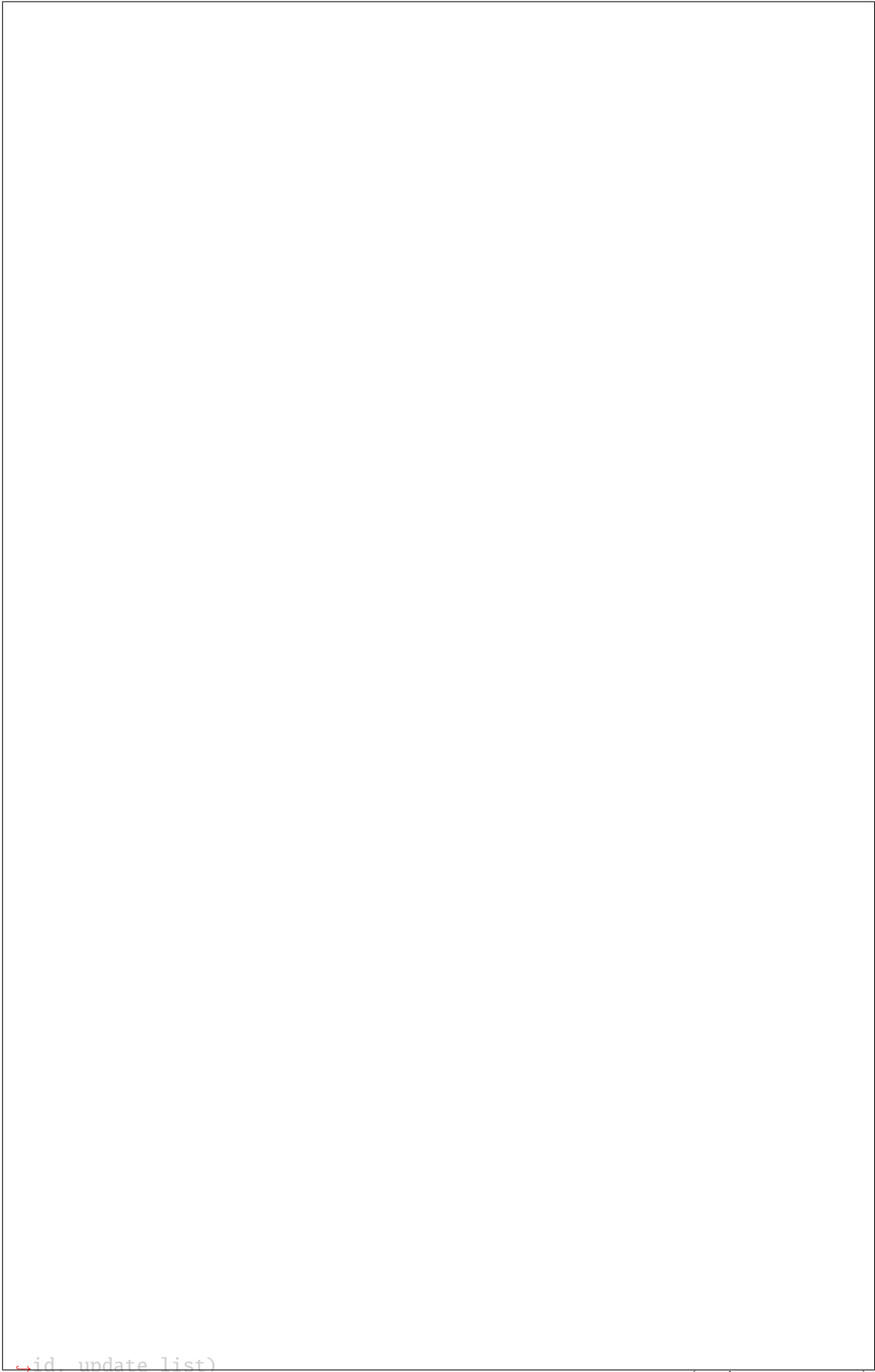
(continued from previous page)



`↔id, create_list)`

(continues on next page)

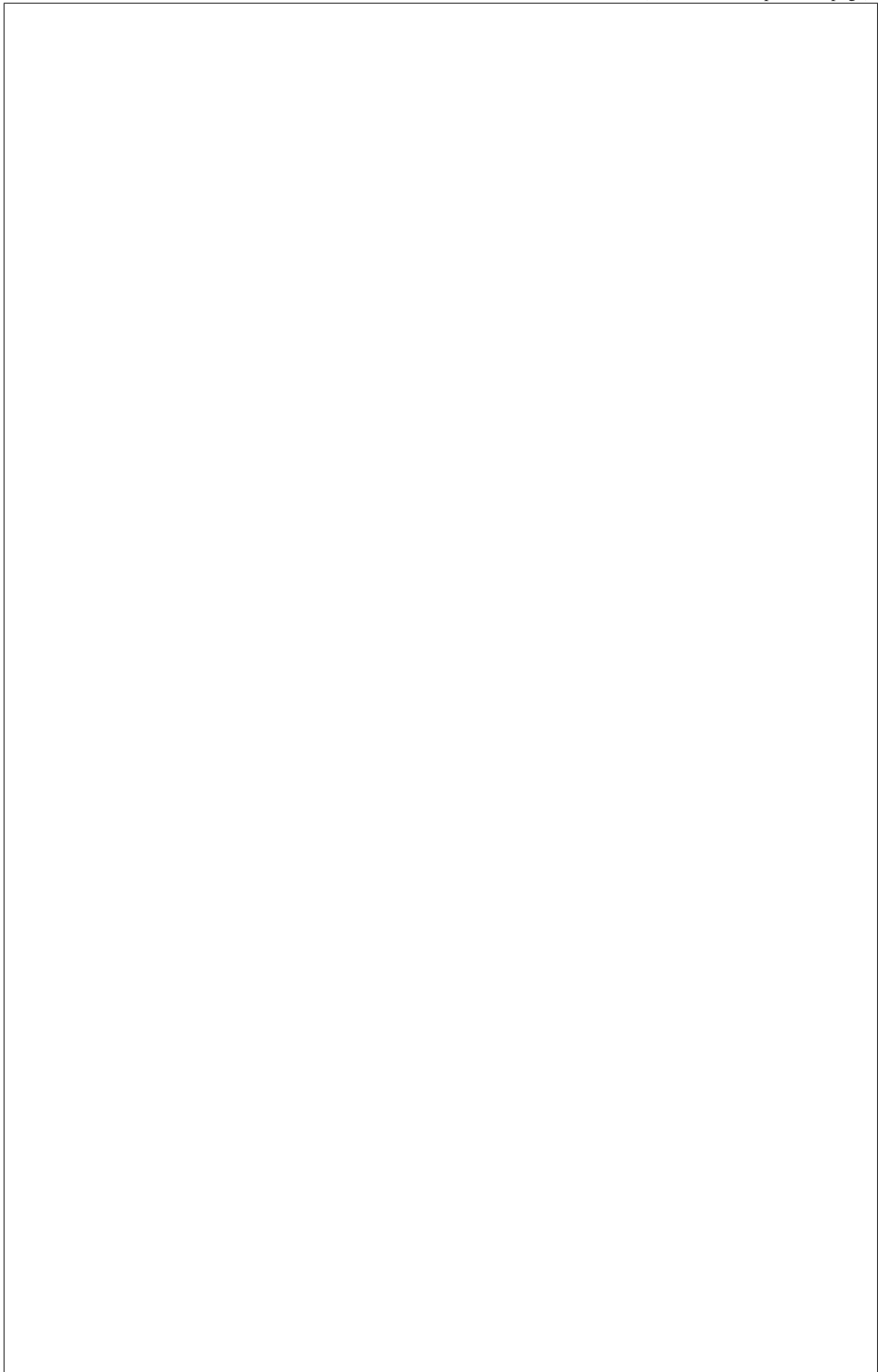
(continued from previous page)



`↪ id, update_list)`

(continues on next page)

(continued from previous page)



(continues on next page)

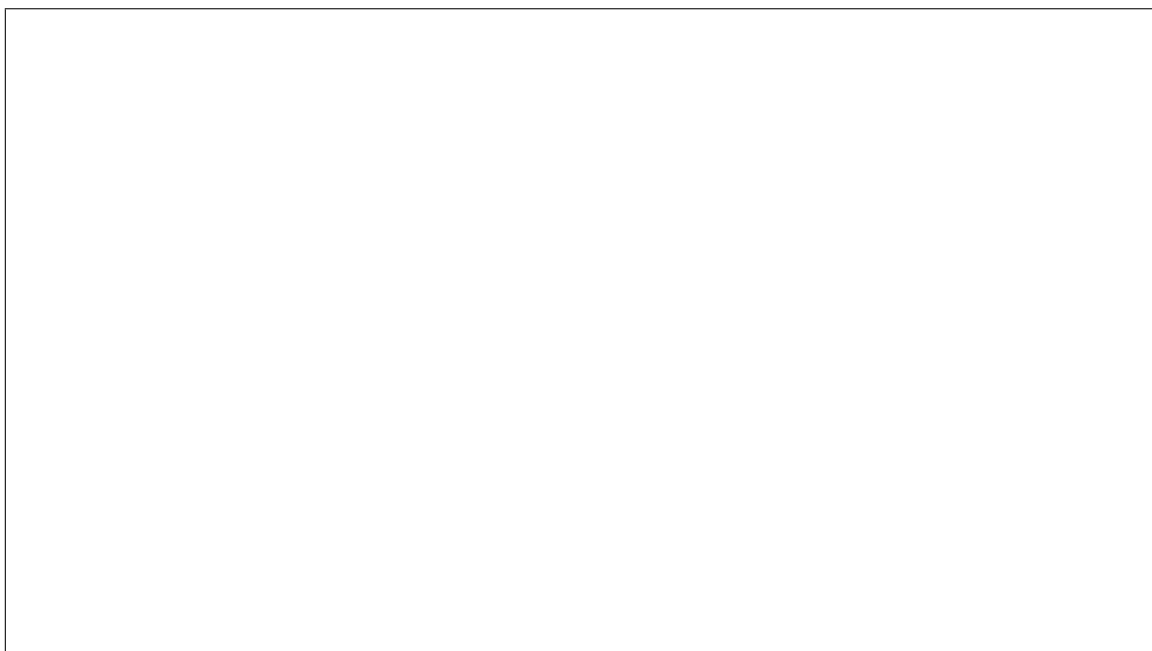
(continued from previous page)

```
↪id, delete_names)
```

Note
dri
cli
is
ven-
dor
spe-
cific
li-
brary
to
con-

ware, for example: python-dracclient, sushy.

resets BIOS settings to factory default on the given node. It calls `cache_bios_settings` automatically to update existing `bios_settings` table once successfully executed.



(continues on next page)

trol
and
man
age
the
bare
meta
hard

- Im-
ple-
men
a
meth
nam
fac
This
meth
need
to
use
the
cle
dec-
o-
ra-
tor.
It

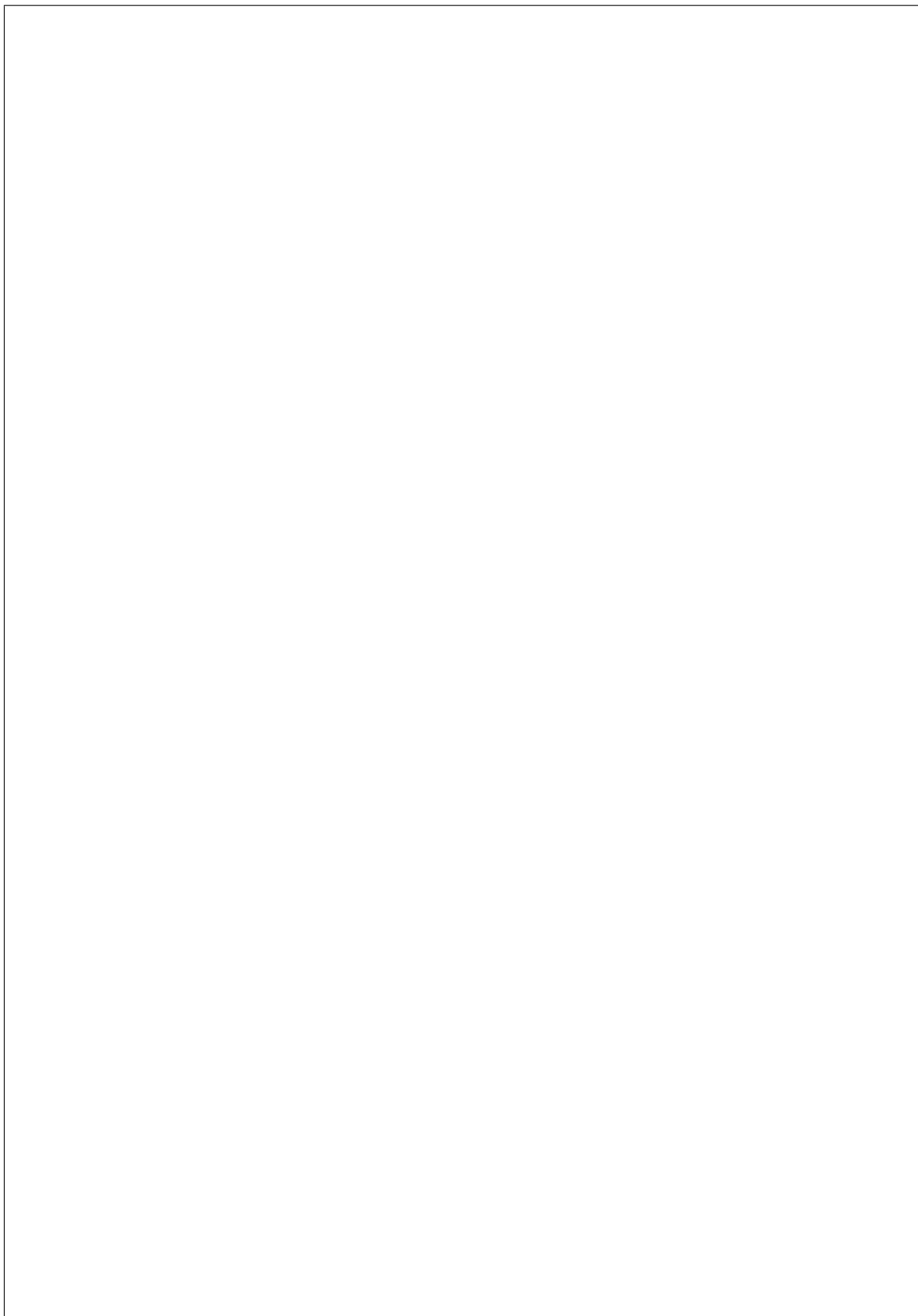
(continued from previous page)



- Im-
ple-
men
a
meth
nam
app
This
meth
need
to
use
the
clear
dec-
o-
ra-

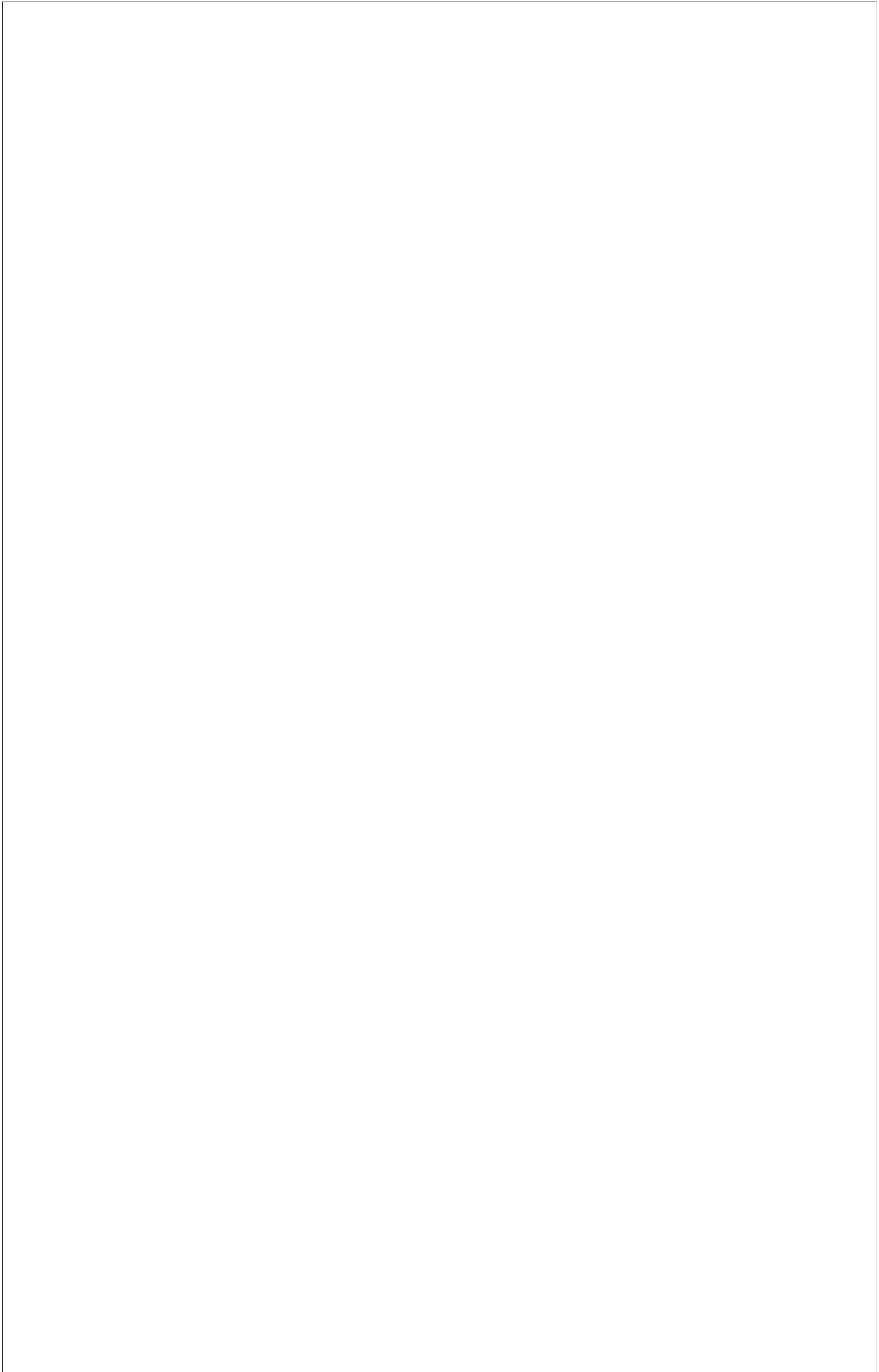
tor.
It

takes the given BIOS settings and applies them on the node. It also calls `cache_bios_settings` automatically to update existing `bios_settings` table after successfully applying given settings on the node.



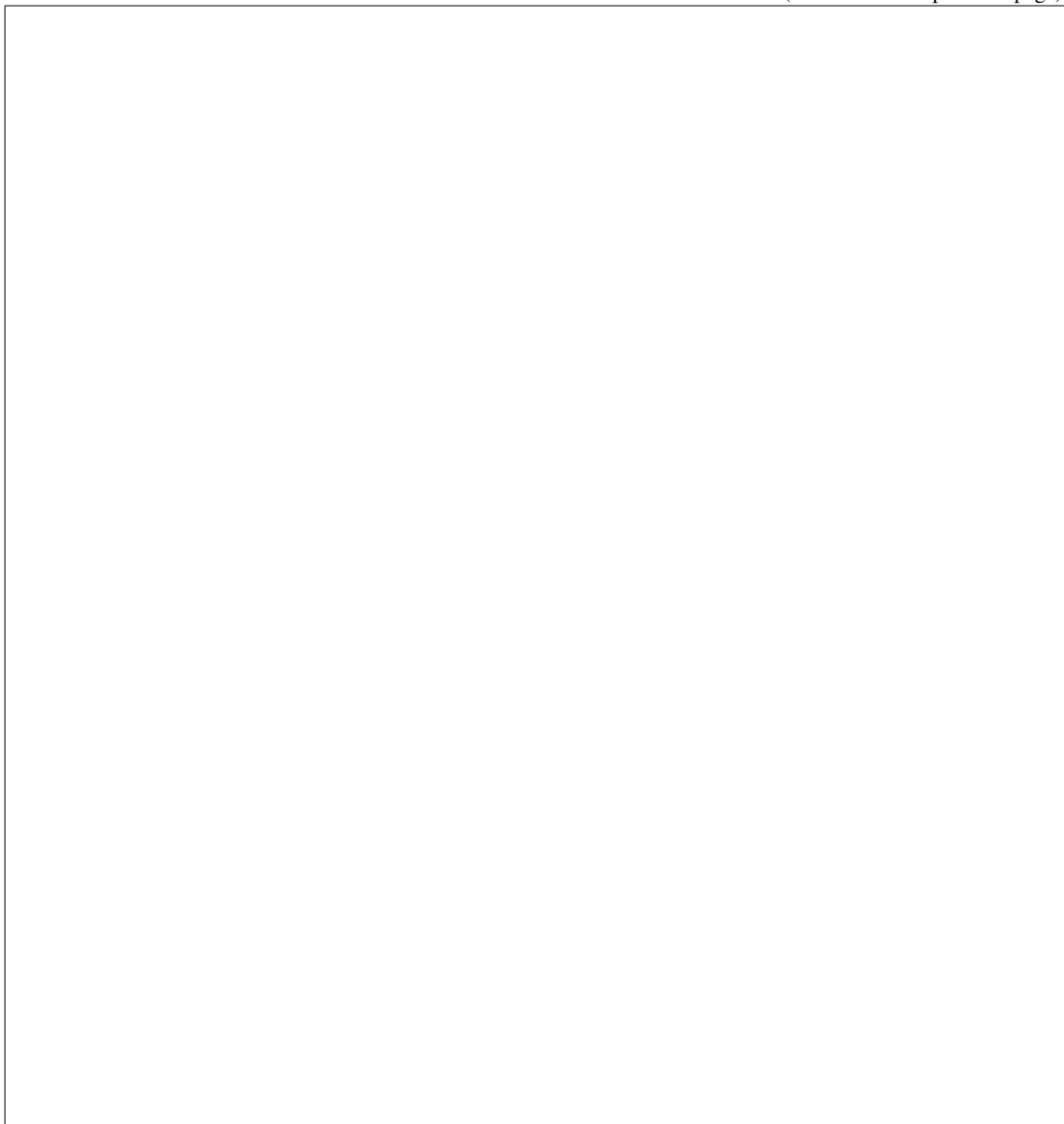
(continues on next page)

(continued from previous page)



(continues on next page)

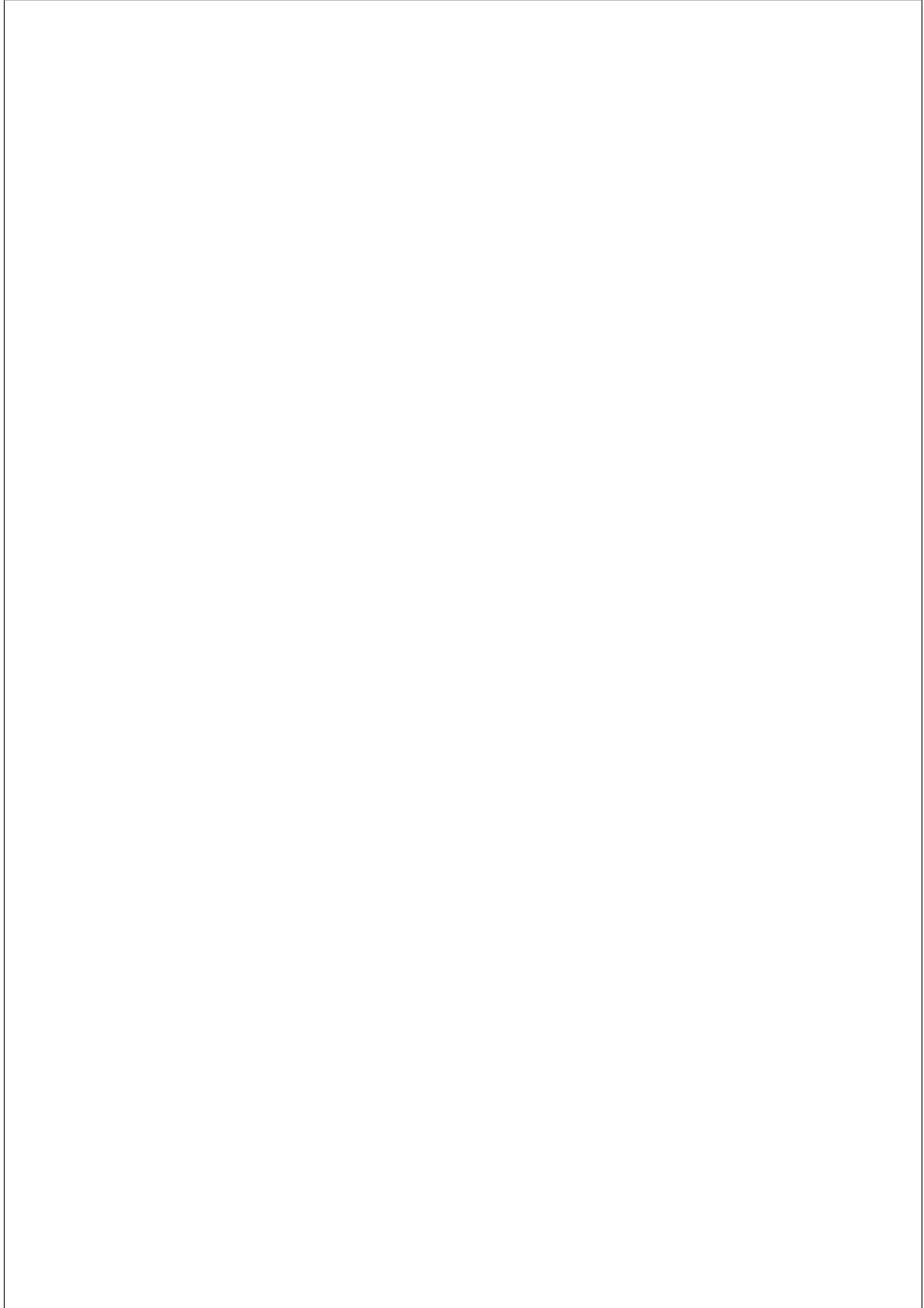
(continued from previous page)



The
set
pa-
ram-
e-
ter
is
a
list
of
BIO
set-
tings
to
be
con-

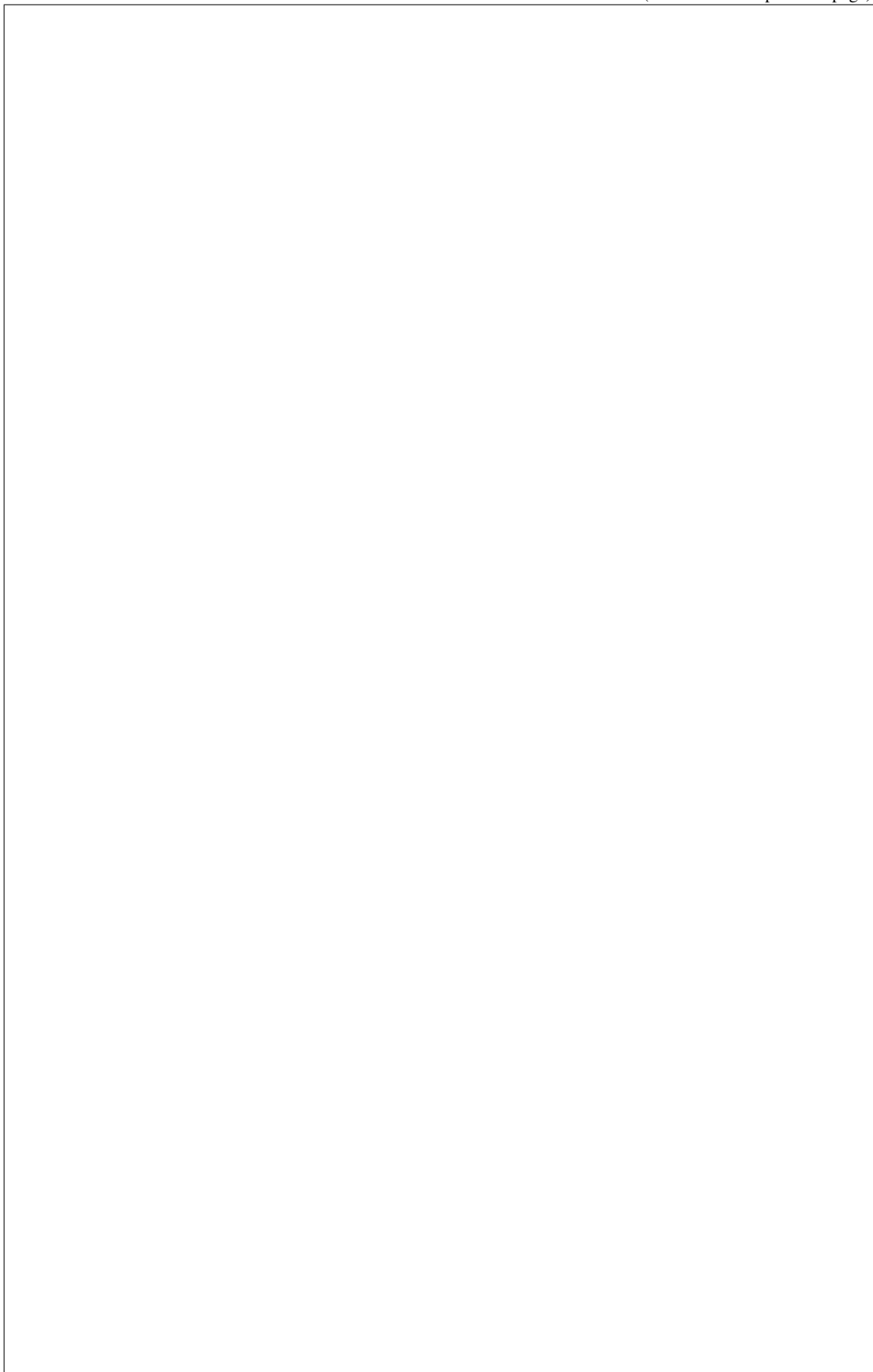
fig-
ured
for
ex-

ample:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Third Party Continuous Integration

Note: This document is a work-in-progress. Unfilled sections will be worked in follow-up patchsets. This version is to get a basic outline and index done so that we can then build on it. (krtaylor)

up their continuous integration test systems.

This
doc-
u-
men-
pro-
vide
tips
and
guid
lines
for
thir
part
driv
de-
vel-
op-
ers
set-
ting

CI Architecture Overview

Requirements Cookbook

Sizing

Infrastructure

ironic job.

jenkins changes

nodepool changes

neutron changes

pre-test hook

cleanup hook

Ironic

Hardware Pool Management

This
sec-
tion
de-
scrib
wha
chan
you
need
to
mak
to
a
your
CI
sys-
tem
to
add
an

Problem

the problem of two jobs trying to use the name target arises. If you have one target machine and a maximum number of one jobs running on your ironic pipeline at a time, then you wont run into this problem. However, one target may not handle the load of ironics daily patch submissions.

Solutions

Zuul v3

Molten Iron

If
you
are
us-
ing
ac-
tual
hard
ware
as
tar-
get
ma-
chin
for
your
CI
test-
ing
then

mol
iron
is
a
tool
that
al-
lows
you
to
re-
serv
hard
ware
from
a
pool

minute to use in your job. Once finished testing, you can unreserve the hardware making it available for the next test job.

Tips and Tricks

Optimize Run Time

Image Server

Other References

Developing a new Deploy Step

class and use the decorator `deploy_step` defined in `ironic/drivers/base.py`. For example, we will implement a `do_nothing` deploy step in the `AgentDeploy` class.



(continues on next page)

at
the
last

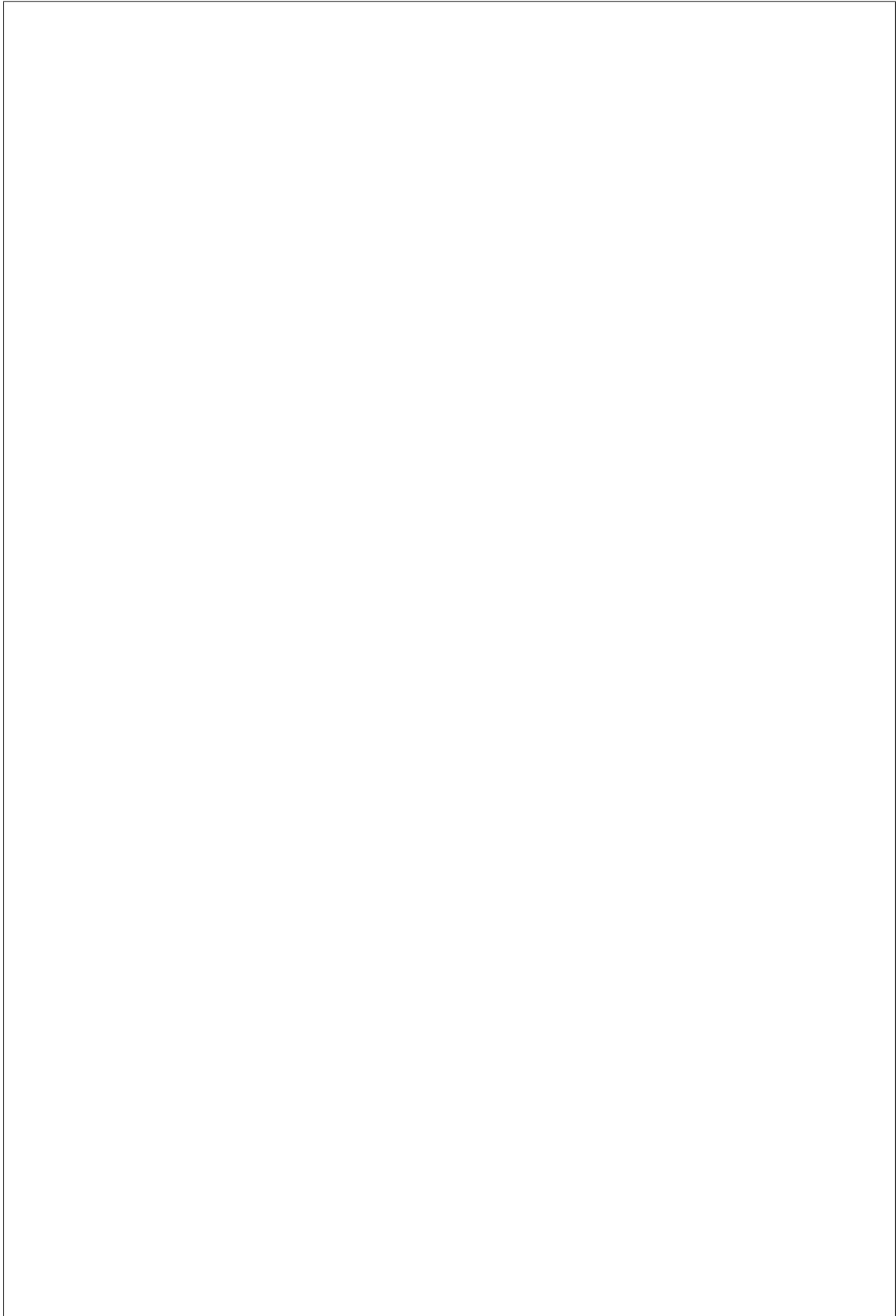
To
sup-
port
cus-
tomi
de-
ploy
men
step
im-
ple-
men
a
new
meth
in
an
in-
ter-
face

from
→ i.
→ in
→ a

cla
→ A
→ A

↳
↳
↳
↳ @
↳ d
↳ s

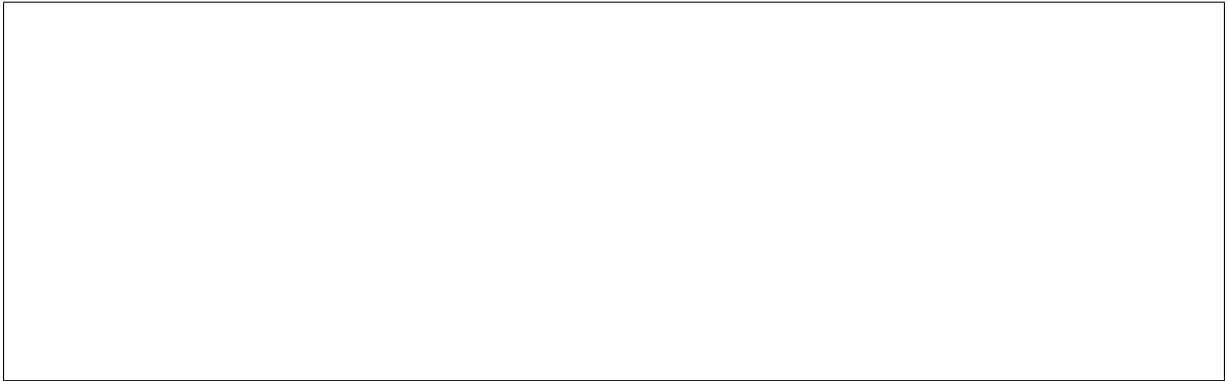
(continued from previous page)



↪test argument."

(continues on next page)

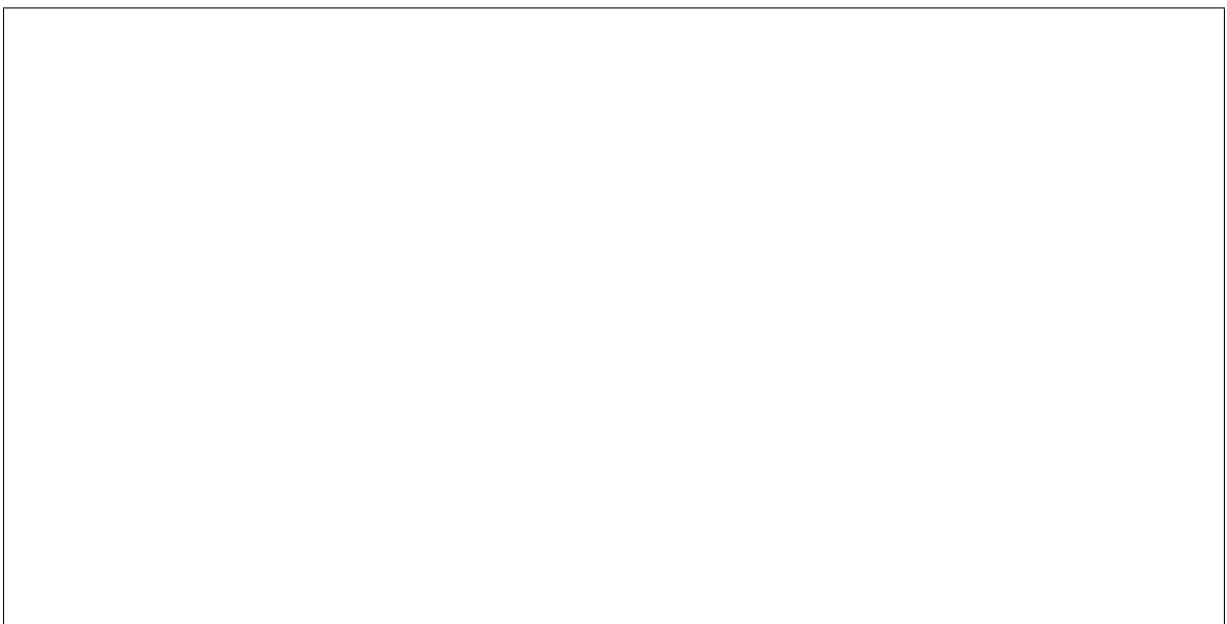
(continued from previous page)



↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳

If
you
wan
to
com
plet
re-
plac
the
de-
ploy
men
pro-
ce-
dure
but
still
have
the
agen

up and running, inherit CustomAgentDeploy:



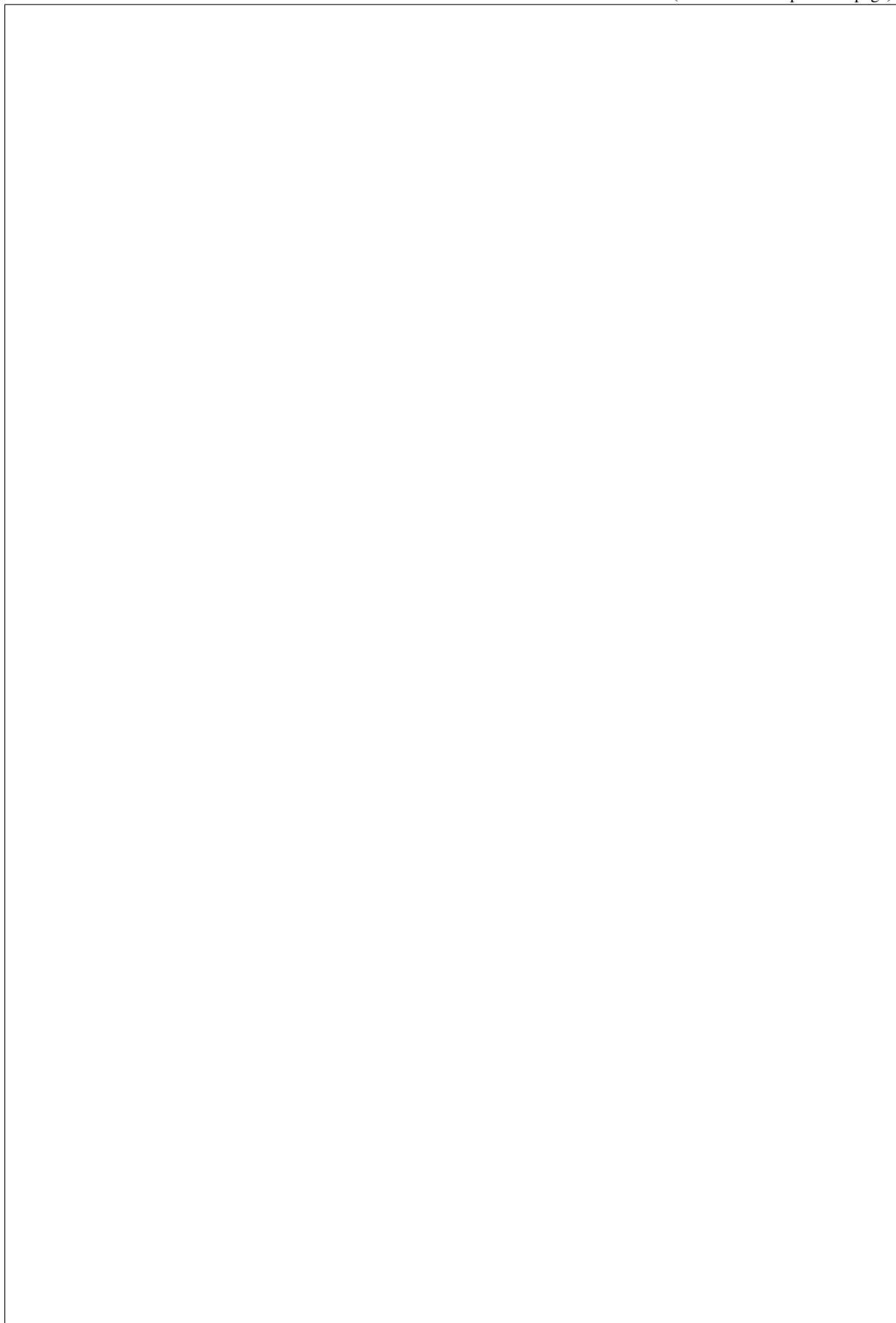
from
↳
↳
↳
↳

clas
↳
↳

↳
↳
↳
↳
↳
↳
↳
↳

(continues on next page)

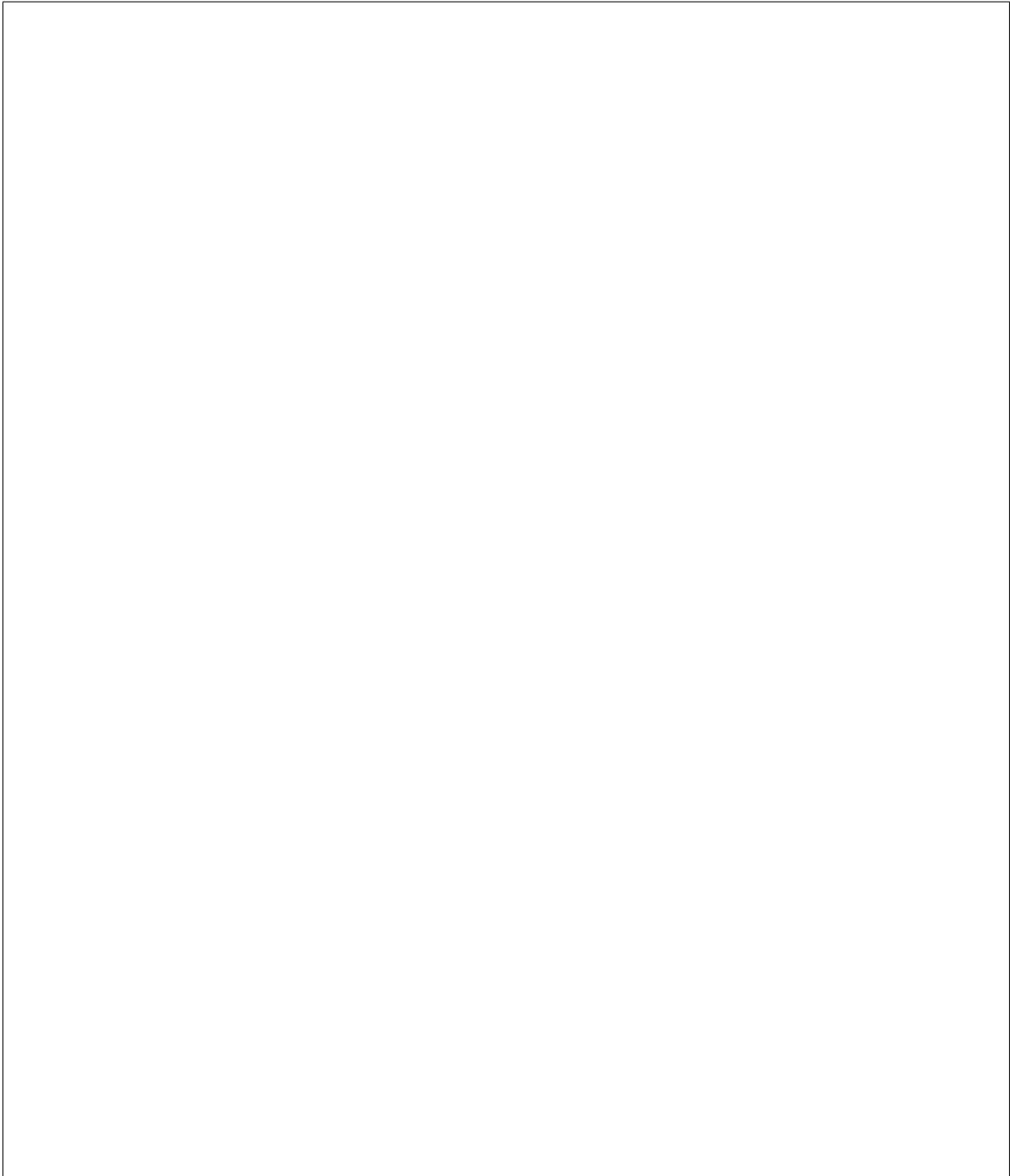
(continued from previous page)



(continues on next page)

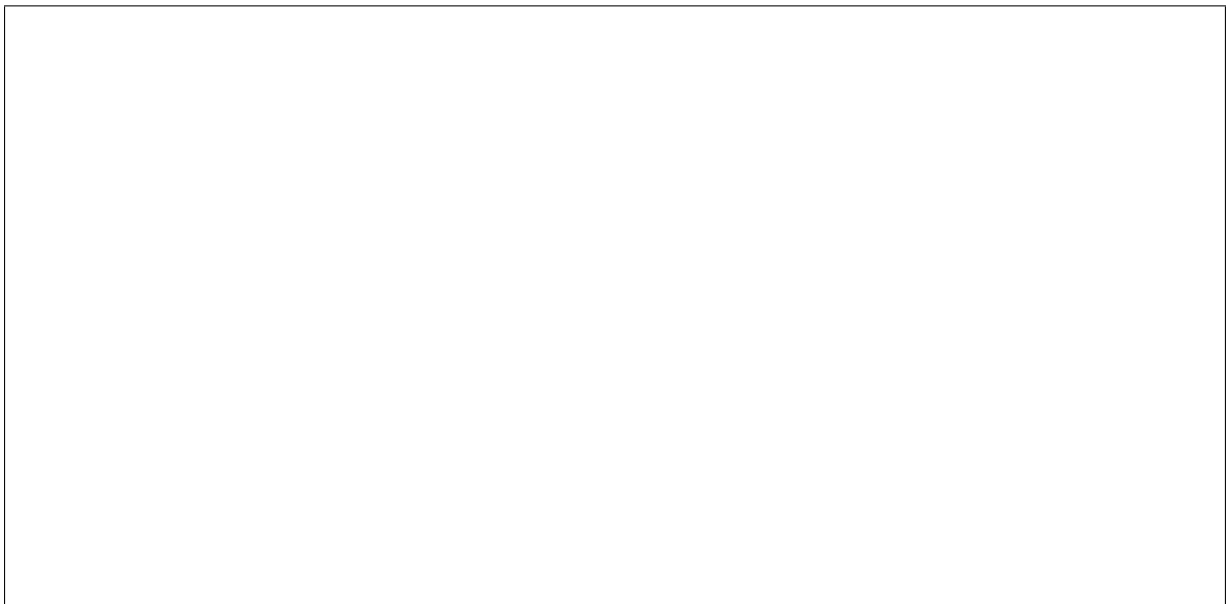
U
→ U
→ U
→ U
→ U
→ U
→ U
→ S
→ V
U
→ U
→ U
→ U
→ U
→ U
→ U
→ U
→ U
→ #
→ .
→ .
→ .
→ C
→ V
U
→ U
→ U
→ U
→ d
→ s
U
→ U
→ U
→ U
→ d
→ m
→ w
→ i
→ U
→ t
→ U
→ *
U
→ U
→ U
→ U
→ U
→ U
→ p

(continued from previous page)



↳
↳
↳
↳
↳@
↳d
↳s
↳
↳
↳
↳
↳d
↳m
↳c
↳b
↳
↳t
↳
↳*
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳p
↳
↳
↳#
↳.
↳.
↳.
↳
↳c
↳b
↳c

Af-
ter
de-
ploy
men
of
the
bare
node
chec



(continues on next page)

the
up-
date
de-
ploy
step

bar
→ n
→ sl
→
→ \$
→ i
→ -
→ f
→ j
→ -
→ c
→ d
→ i
→ i

The
above
com
man
out-
puts
the
dri
as
fol-
low-
ing:

→ {
↳
→ ↳
→
→ "
→ i
→ i
→ "
→
→ {
↳
→ ↳
→ ↳
→ ↳
→ .
→ .
→ .
→

(continued from previous page)



(continues on next page)

(continued from previous page)

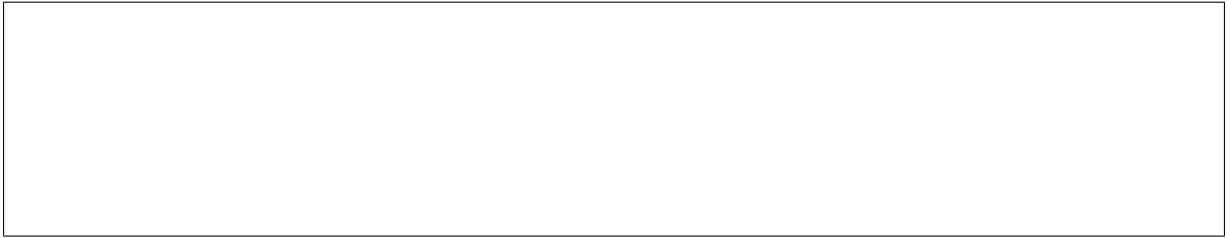


↔

↔`"This is a test argument."`

(continues on next page)

(continued from previous page)



Note: Similarly, clean steps can be implemented using the `clean_step` decorator.

mented in a custom [IPA hardware manager](#). All in-band deploy steps must have priorities between 41 and 99, see *Agent steps* for details.

↳
↳
↳
↳
}

In-
band
de-
ploy
step
(de-
ploy
step
that
are
run
in-
side
the
range
have
to
be
im-
ple-

8.1.
Net
wor
In-
te-
gra
tion

In
or-
der
to
test
the
in-
te-

vices, support has been added to `devstack` to mimic an external physical switch. Here we include a recommended configuration for `devstack` to bring up this environment.

Ironic multitenant networking and DevStack

Using VMs as baremetal servers

gra-
tion
be-
twee
the
Bare
Met
and
Net-
worl
ing
ser-

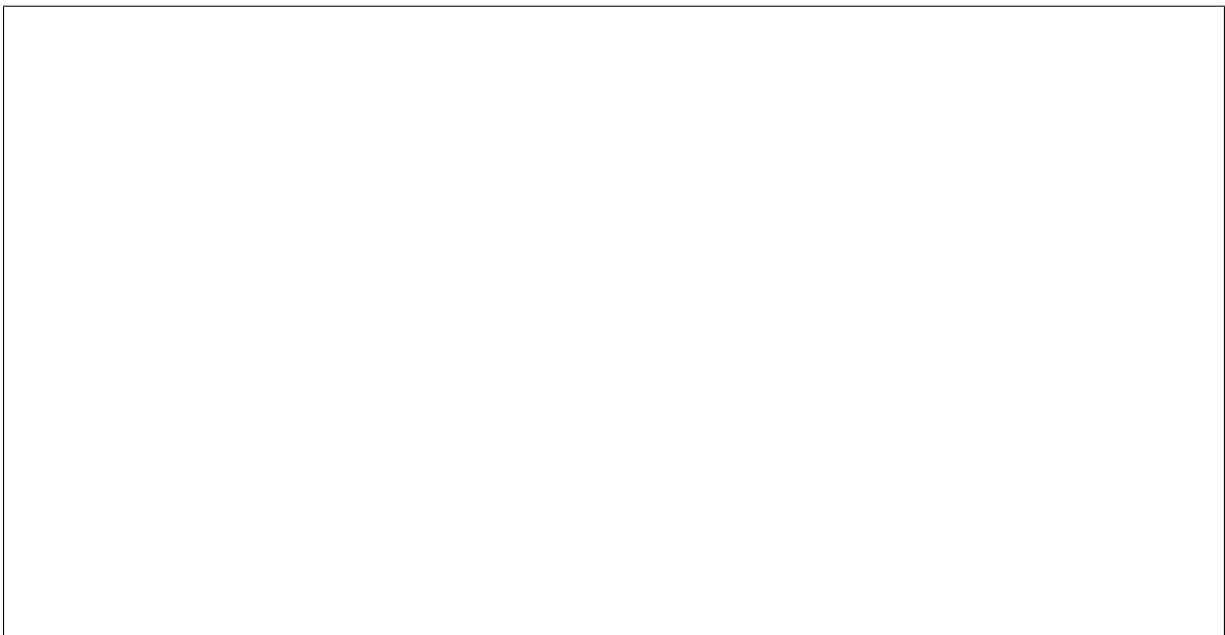
This
guid
will
walk
you
thro
us-
ing
Ope
Stac
Iron
with
the
ML2
net
plu-
gin.

This
sce-
nari
show
how
to
setu
De-
vs-
tack
to
use
Iron

baremetal servers and ML2 networking-generic-switch that interacts with OVS.

DevStack Configuration

tered in ironic. networking-generic-switch driver will be installed and configured in Neutron.



(continues on next page)

in-
te-
gra-
tion
with
VM
as

The
fol-
low-
ing
is
loc-
con-
that
will
setu-
De-
vs-
tack
with
3
VM
that
are
reg-
is-

[[1

↪ #
↪ C
↪ i
↪ f
↪ i
↪ d
↪ p
↪
enal
↪ p
↪ i
↪ h
↪ /
↪ o
↪ o
↪ o
↪ i

(continued from previous page)



(continues on next page)

↪ #
↪ I
↪ n
↪ g
↪ S
↪ N
↪ M
↪ d
↪ t
↪ i
↪ w
↪ O
enab
↪ p
↪ n
↪ g
↪ s
↪ h
↪ /
↪ o
↪ o
↪ o
↪ n
↪ g
↪ s

↪ #
↪ A
↪ l
↪ l
↪ i
↪ w
↪ r
↪ I
↪ n
IRO
↪ U
↪ L
↪ L

IRO
↪ E
↪ N
↪ I
↪ n
IRO
↪ N
↪ I

(continued from previous page)



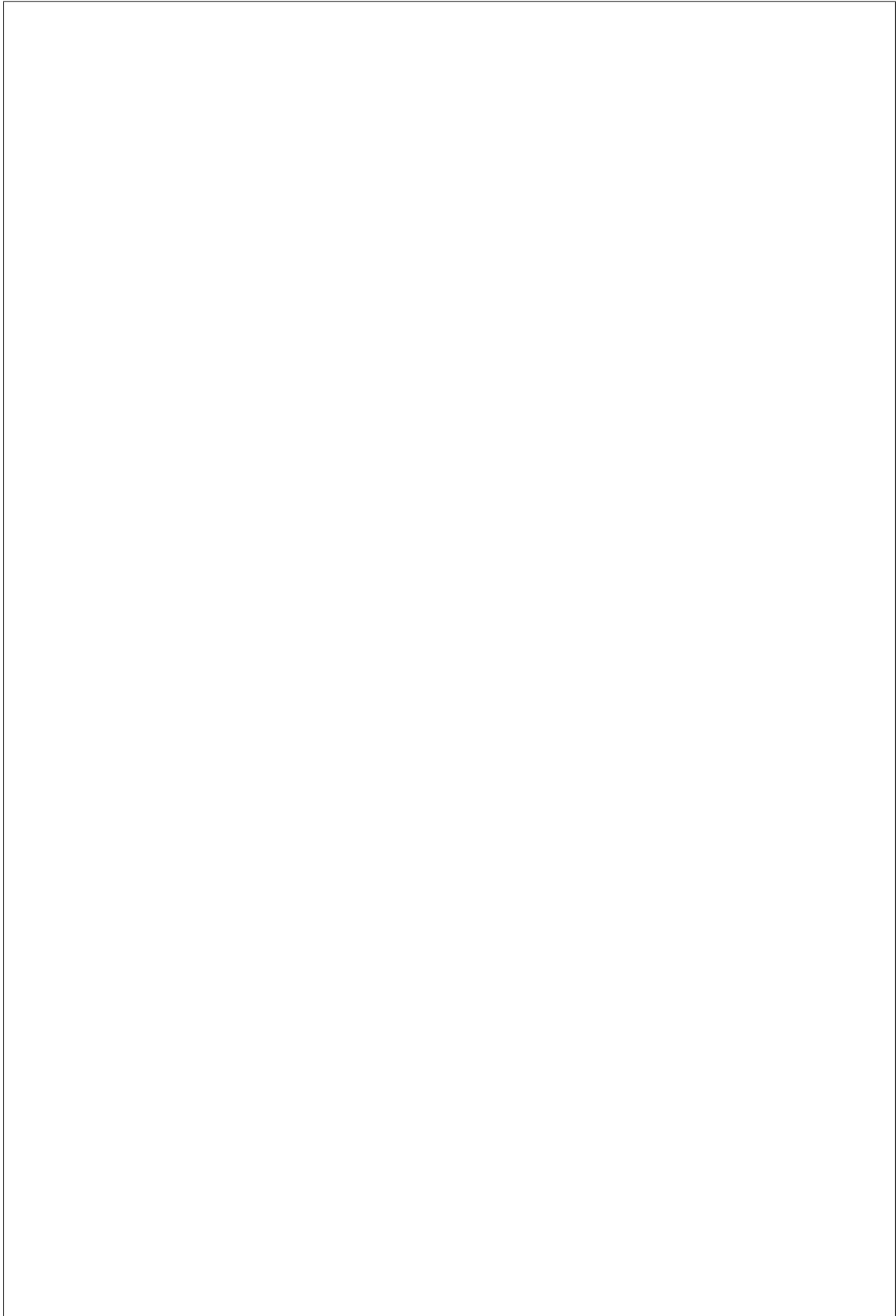
(continues on next page)

→ #
→ C
OVS.
→ P
→ B
PHY
→ N
IRO
→ P
→ N
→ N
→ p
IRO
→ P
→ S
→ P
→ 0
→ 5
→ 0
→ 2
IRO
→ P
→ S
→ G
→ 0
→ 5
→ 1

Q_
→ P
ENAB
→ T
→ V
Q_
→ M
→ T
→ N
→ T
TEN
→ V
→ R

→ #
→ C
ADM
→ P

(continued from previous page)



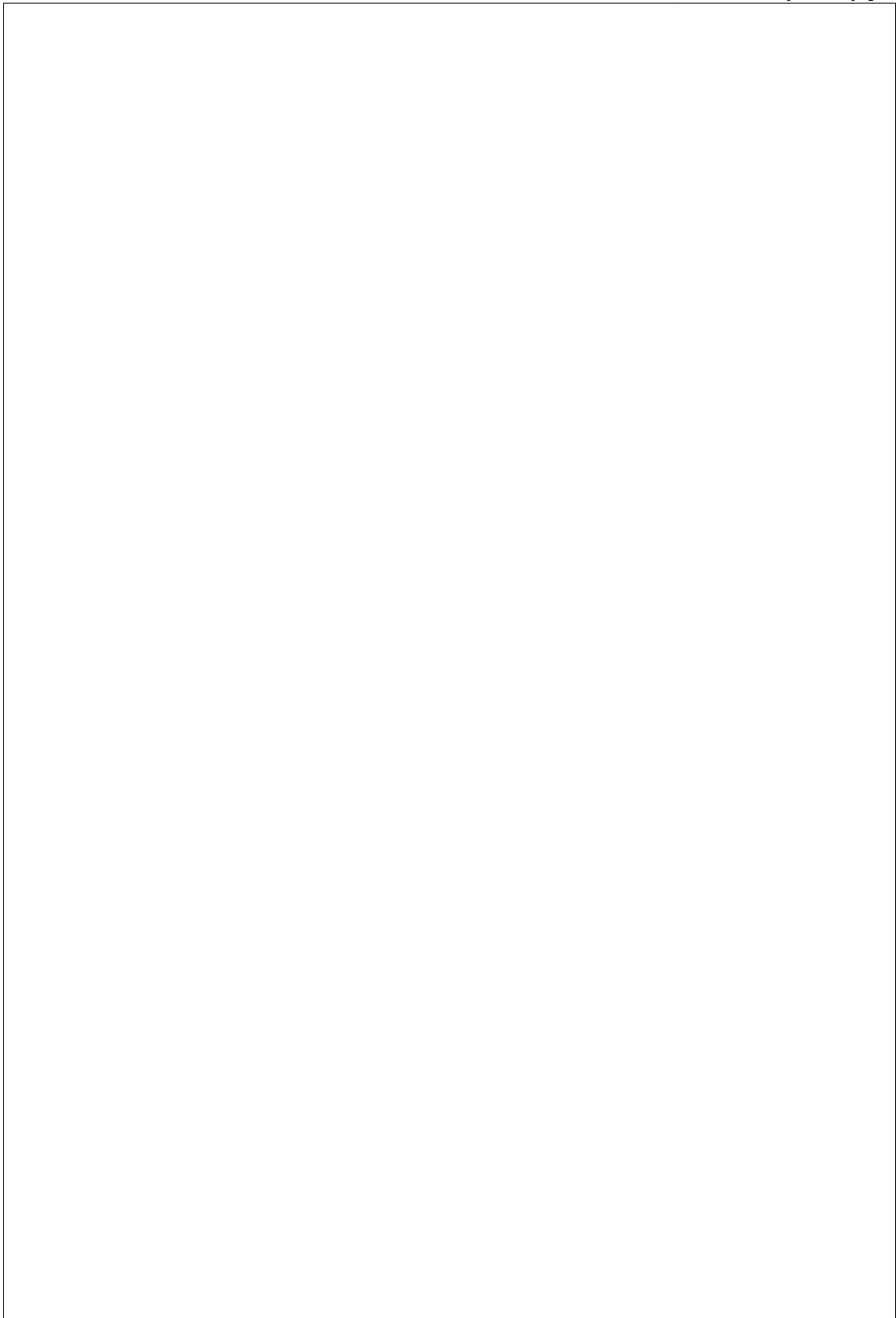
(continues on next page)

RAB
→ P
DAT
→ P
SER
→ P
SER
→ T
SWI
→ H
SWI
→ T
→ K

→ #
→ E
→ I
→ A
→ a
→ I
→ C
enal
→ s
→ i
enal
→ s
→ i
→ a
enal
→ s
→ i
→ c

→ #
→ D
→ n
→ n
→ s
→ L
→ i
→ d
→ n
→ s
→ i
→ a
→
dis
→ s
→ n
→ n

(continued from previous page)



(continues on next page)

↪ #
↪ E
↪ S
↪ f
↪ t
↪ d
↪ d
↪ i
↪
enab
↪ S
↪ S
↪ p
enab
↪ S
↪ S
↪ o
enab
↪ S
↪ S
↪ c
enab
↪ S
↪ S
↪ a

↪ #
↪ D
↪ H
dis
↪ s
↪ h

↪ #
↪ D
↪ C
dis
↪ s
↪ c
↪ c
↪ S
↪ c
↪ a
↪ c
↪ v

(continued from previous page)



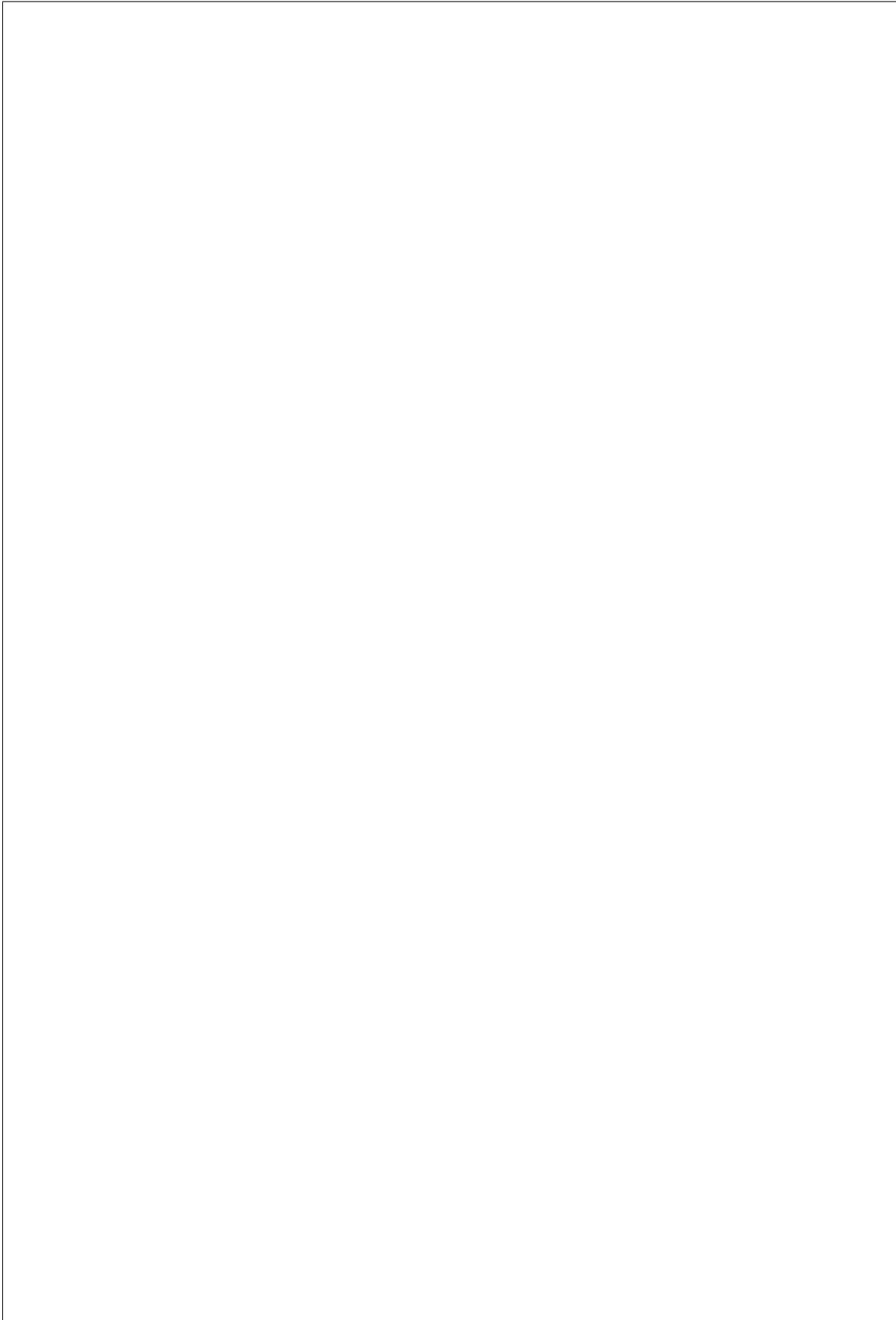
(continues on next page)

→ #
→ D
→ T
dis
→ S
→ t

→ #
→ S
→ t
→ t
→ U
→ '
→ a
→ r
→ f
→ t
→ d
→ d
→ i
→
SWI
→ E
→ T

→ #
→ C
→ 3
→ v
→ m
→ t
→ p
→ a
→ I
→ '
→ b
→ n
→
IRO
→ V
→ C
IRO
→ B
→ B
→ O

(continued from previous page)



(continues on next page)

↪ #
↪ E
↪ a
↪ h
↪ t
↪
↪ i
↪ n
↪

↪ #
↪ E
↪ H
↪ T
↪ f
↪ h

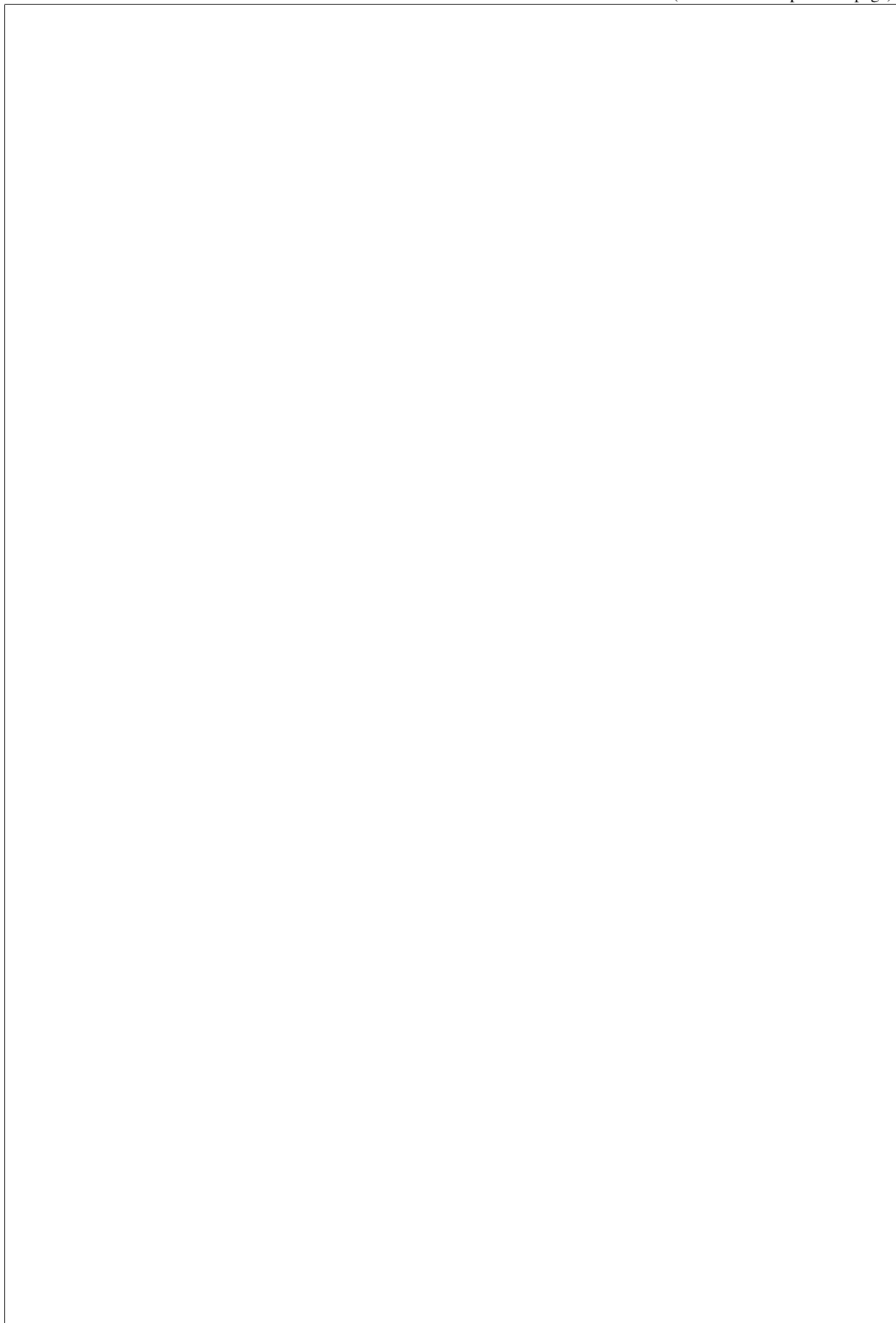
↪ #
↪ D
↪ '
↪ f
↪ t
↪ m
↪ h
↪ t
↪ r
↪ e
↪ o
↪ a

↪ #
↪ i
↪
↪ m
↪ o
↪ p
↪ a
↪ m

↪ #
↪ E
↪ M
↪ I
↪ f

↪ #
↪ E
↪ P
↪ I
↪ f

(continued from previous page)



(continues on next page)

↪ #
↪ D
↪ D
↪ I
↪ #
↪ C
↪ t
↪ t
↪ a
↪ t
↪ d
↪ d
↪ f
↪ n
↪ c
↪ b
↪ d
↪
↪ #
↪ T
↪ d
↪ s
↪ b
↪ i
↪ t
↪ e
↪ l
↪ a
↪
IRON
↪ D
↪ D
↪ #
↪ T
↪ p
↪ b
↪ r
↪ t
↪ m
↪ p
↪ v
↪ t
↪ c

(continued from previous page)



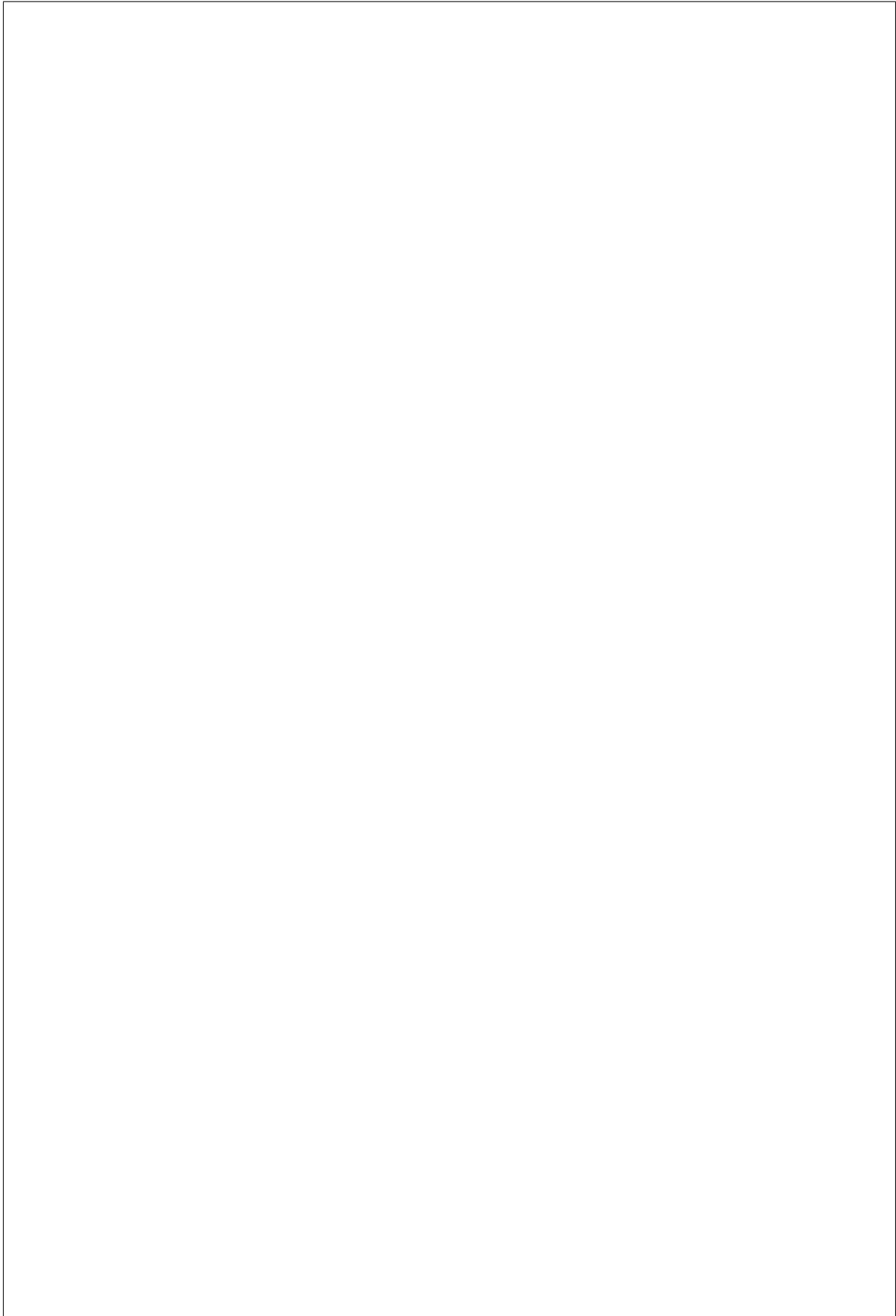
(continues on next page)

↪ #
↪ f
↪ n
↪
IRON
↪ V
↪ S
↪ R
IRON
↪ V
↪ S
↪ D

↪ #
↪ S
↪ o
↪ t
↪ e
↪ p
↪ i
↪ G
↪
↪ U
↪ 0
↪ f
↪ n
↪ e
↪ p
↪
IRON
↪ V
↪ E
↪ D

↪ #
↪ T
↪ b
↪ y
↪ o
↪ I
↪ r
↪ f
↪ s
↪
↪ s
↪ t
↪ t
↪ T

(continued from previous page)



(continues on next page)

IRO
→ B
→ D
→ R

VIR
→ D

→ #
→ B
→ d
→ L
→ D
→ c
→ a
→ 1
→ 0
→ 0
→ 0
→ 2
→ n
→ f
→ i
→

→ #
→ I
→ t
→ o
→ w
→ t
→ h
→ n
→ L
→ y
→ m
→ a
→ w
→ t

→ #
→ f
→
NET
→ G
→ 1
→ 0
→ 1

(continued from previous page)



FIX

→ R

→ 1

→ 0

→ 0

→ 2

FIX

→ N

→ S

→ #

→ L

→ a

→ o

→ t

→ f

LOG

→ \$

→ d

→ l

→ l

LOG

→ \$

→ l

IRO

→ V

→ L

→ D

→ \$

→ i

→ b

→ l

8.1.

Boo

from

Vol

Star

ing

with

the

Pike

re-

leas

it

is

also

pos-

ing booting from Cinder volumes with VMs.

Ironic Boot-from-Volume with DevStack

supported from the Pike release.

si-
ble
to
use
De-
vS-
tack
for
test-

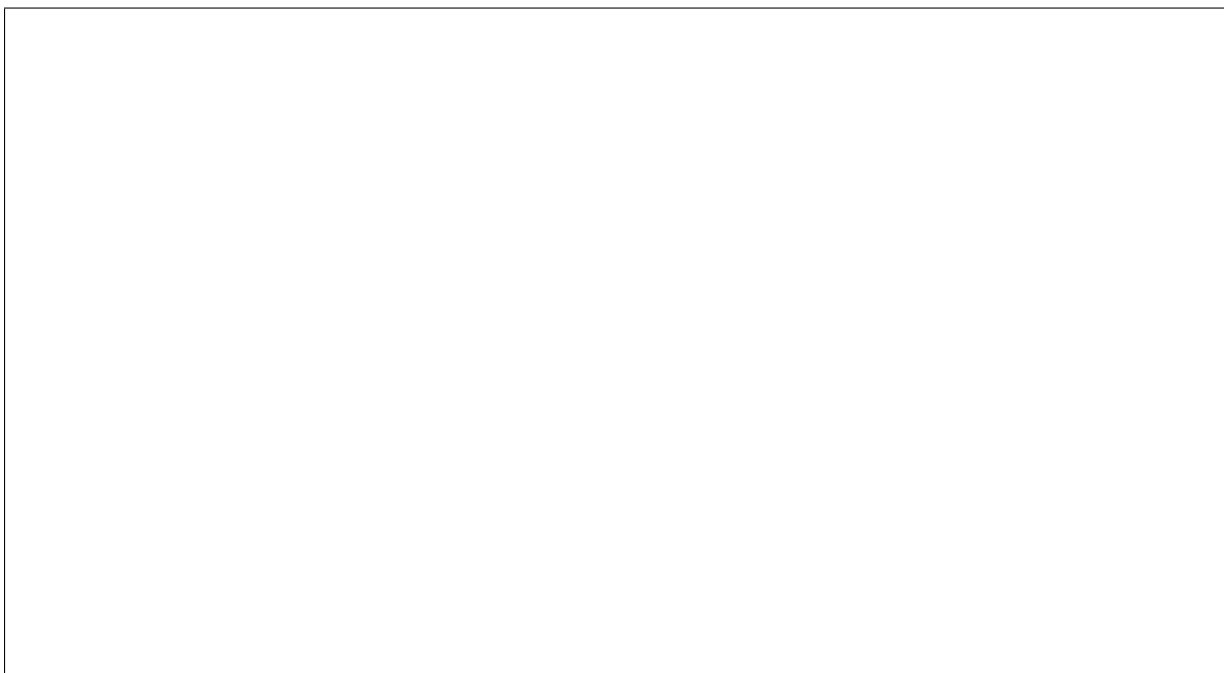
This
guid
show
how
to
setu
De-
vS-
tack
for
en-
ablin
boot
from
volu
fea-
ture,
whic
has
been

This
sce-
nari
show
how
to
setu
De-
vS-
tack
to
en-
able
node
to
boot

aged by cinder with VMs as baremetal servers.

DevStack Configuration

tered in ironic. A volume connector with IQN is created for each node. These connectors can be used to connect volumes created by cinder. The detailed description for DevStack is at [Deploying Ironic with DevStack](#).



(continues on next page)

from
vol-
ume
man

The
fol-
low-
ing
is
loc-
con-
that
will
setu
De-
vS-
tack
with
3
VM.
that
are
reg-
is-

[[1

enal

↪ p

↪ i

↪ h

↪ /

↪ o

↪ o

↪ o

↪ i

IRON

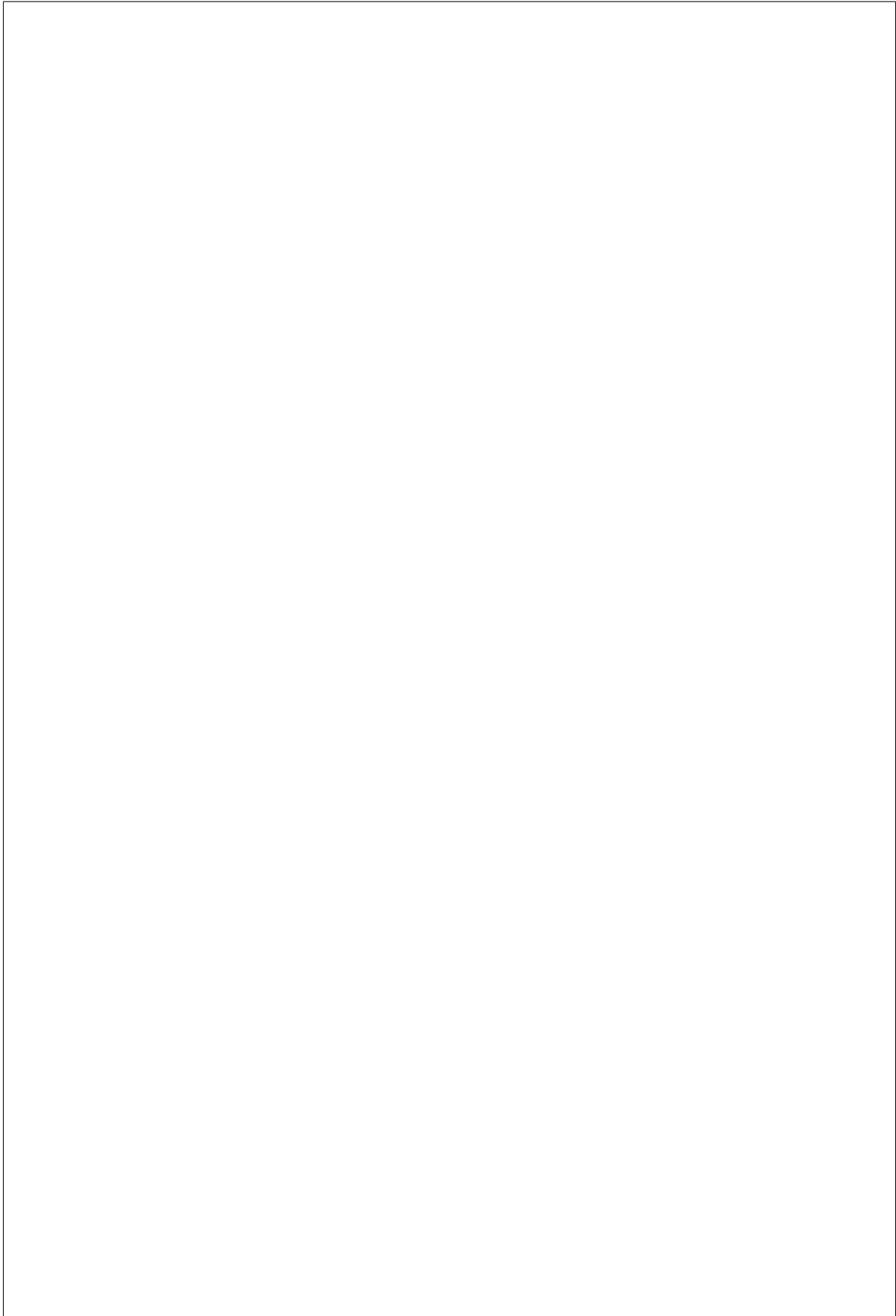
↪ S

↪ I

↪ #

↪ C

(continued from previous page)



(continues on next page)

ADM
→ P
DAT
→ P
RAB
→ P
SER
→ P
SER
→ T
SWI
→ H
SWI
→ T
→ K

→ #
→ E
→ N
→ w
→ i
→ r
→ b
→ I
→ a
→ d
→ n
→ n
→
dis
→ s
→ n
→ n
dis
→ s
→ n
→ n
enal
→ s
→ q
→ s
enal
→ s
→ q
→ a
enal
→ s
→ q
→ d

(continued from previous page)



(continues on next page)

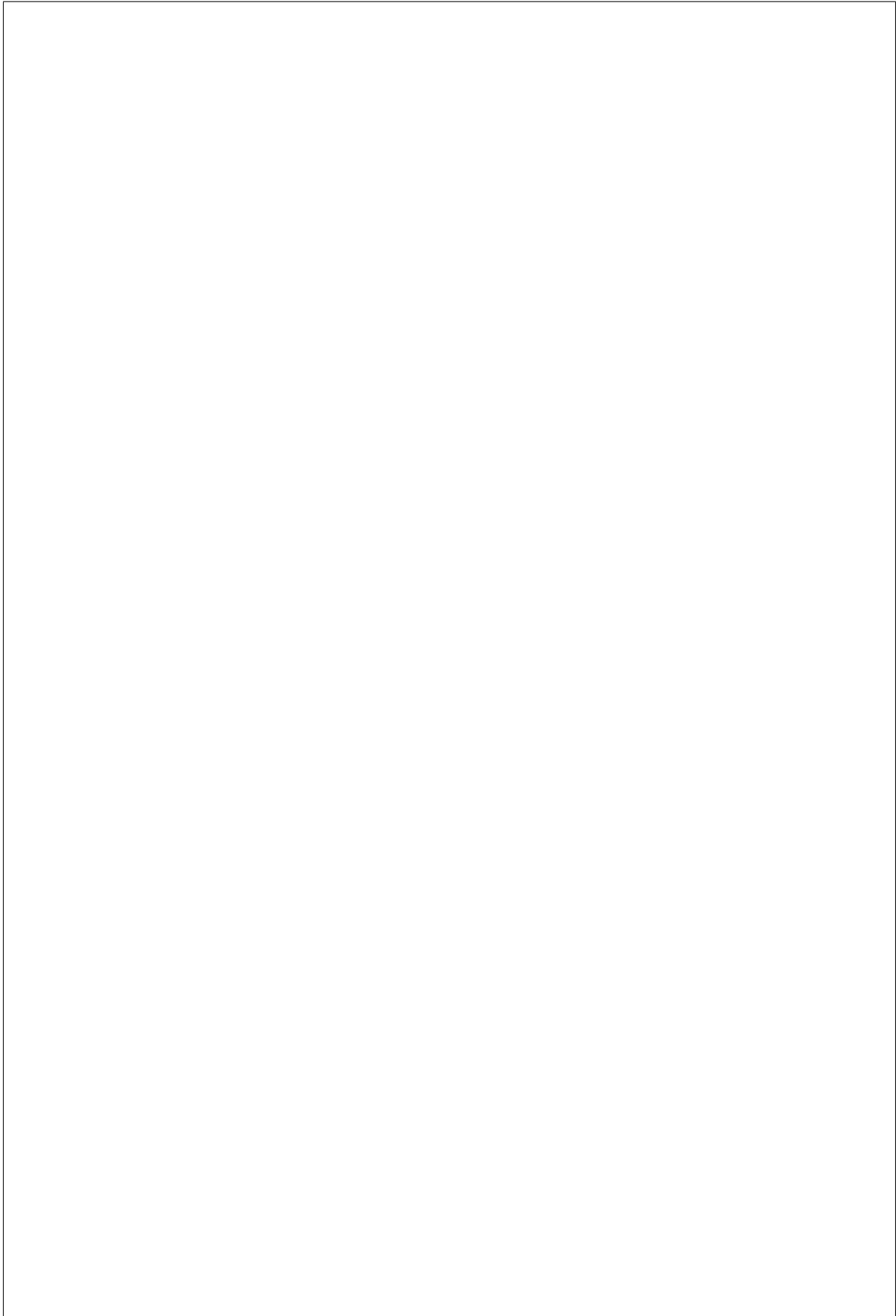
enal
→ s
→ q
→ l
enal
→ s
→ q
→ m
enal
→ s
→ n

→ #
→ E
→ S
→ f
→ t
→ d
→ d
→ i
→
enal
→ s
→ s
→ p
enal
→ s
→ s
→ o
enal
→ s
→ s
→ c
enal
→ s
→ s
→ a

→ #
→ D
→ H
dis
→ s
→ h

→ #
→ D
→ H

(continued from previous page)



(continues on next page)

dis
→ s
→ h
→ h
→ a
→ a
→ h
→ a
→ c
→ h
→ a
→ c
→ h
→ e

→ #
→ S
→ t
→ U
→ '
→ a
→ r
→ f
→ t
→ d
→ d
→ i
→
SWI
→ E
→ T

→ #
→ C
→ 3
→ v
→ m
→ t
→ p
→ a
→ I
→ '
→ b
→ n
→
IRO
→ V
→ C

(continued from previous page)



(continues on next page)

IRO
→ B
→ B
→ O
DEF
→ I
→ T

→ #
→ E
→ a
→ h
→ t
→ L
→ i
→ n
→

→ #
→ E
→ H
→ T
→ f
→ h

→ #
→ D
→ '
→ f
→ t
→ m
→ h
→ t
→ r
→ e
→ o
→ a

→ #
→ i
→ L
→ m
→ o
→ p
→ a
→ m

→ #
→ E
→ M
→ I
→ f

(continued from previous page)



(continues on next page)

↪ #
↪ E
↪ P
↪ I
↪ f

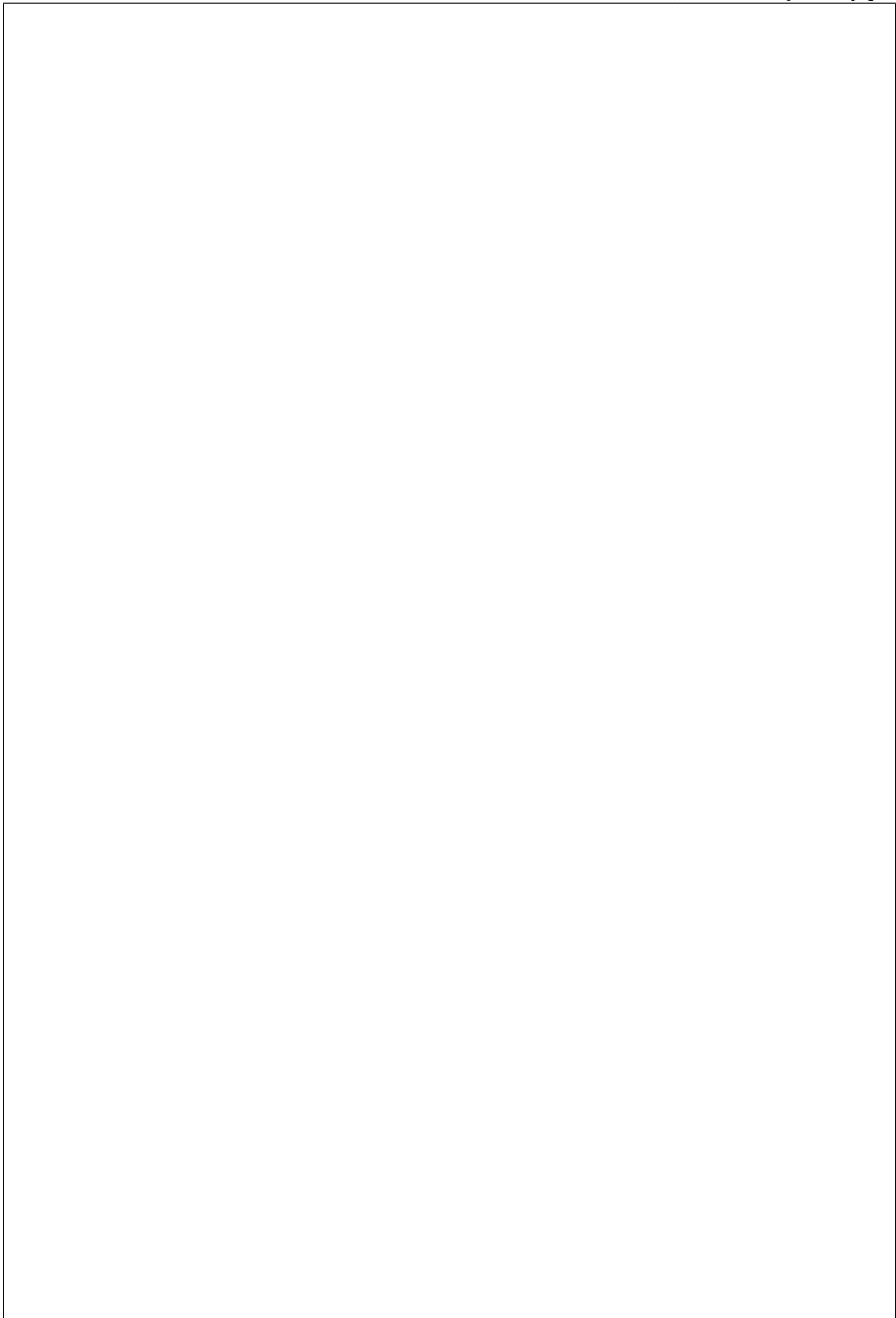
↪ #
↪ D
↪ D
↪ I

↪ #
↪ C
↪ t
↪ t
↪ a
↪ a
↪ t
↪ d
↪ d
↪ f
↪ n
↪ c
↪ b
↪ d
↪

↪ #
↪ T
↪ d
↪ s
↪ b
↪ i
↪ t
↪ e
↪ l
↪ a
↪
IRO
↪ D
↪ D

↪ #
↪ T
↪ p
↪ b
↪ r
↪ t
↪ m
↪ p
↪ v
↪ t
↪ c

(continued from previous page)



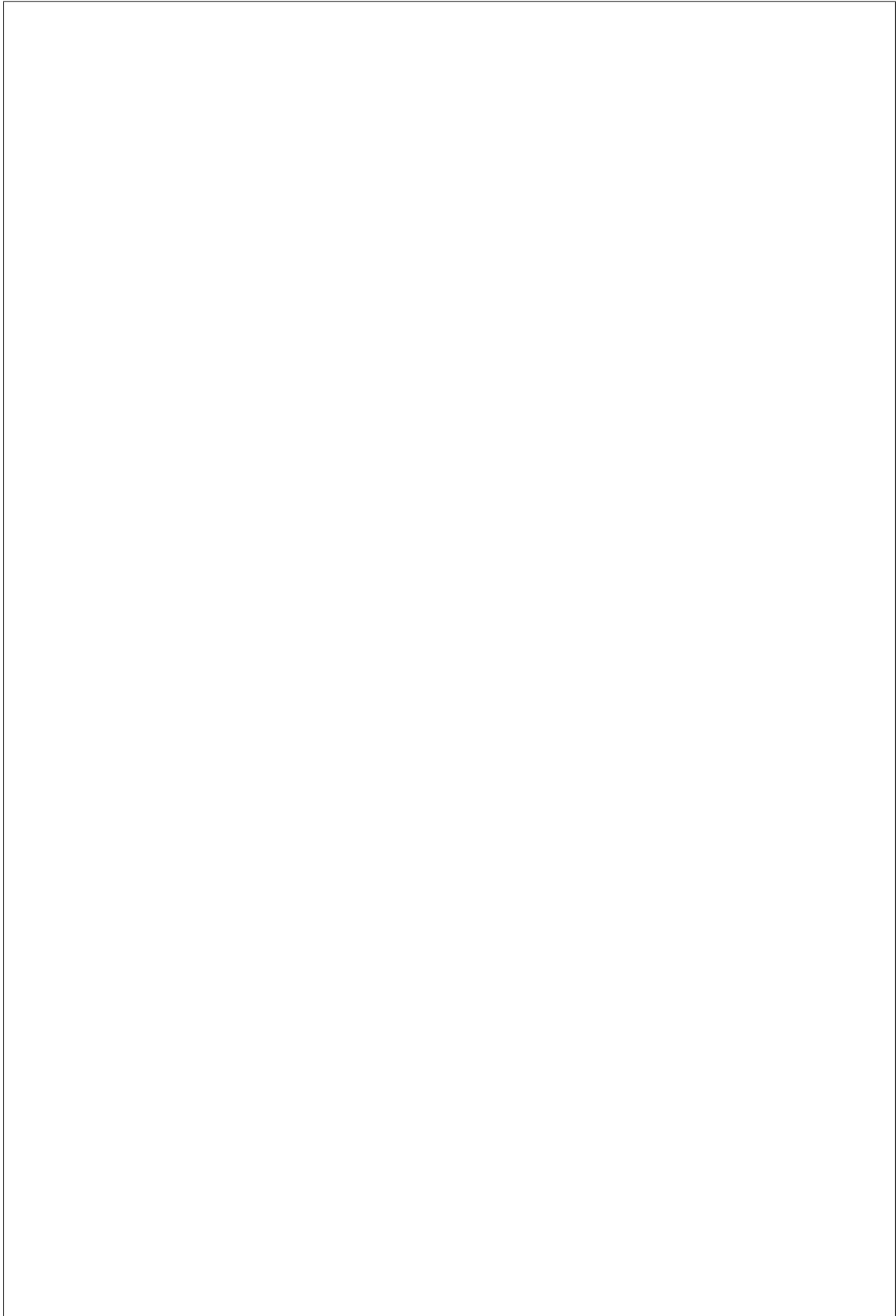
(continues on next page)

↪ #
↪ f
↪ n
↪
IRON
↪ V
↪ S
↪ R
IRON
↪ V
↪ S
↪ D

↪ #
↪ S
↪ o
↪ t
↪ e
↪ p
↪ i
↪ G
↪
↪ U
↪ 0
↪ f
↪ n
↪ e
↪ p
↪
IRON
↪ V
↪ E
↪ D

↪ #
↪ T
↪ b
↪ y
↪ o
↪ I
↪ r
↪ f
↪ s
↪
↪ s
↪ t
↪ t
↪ T

(continued from previous page)



(continues on next page)

IRO
→ B
→ D
→ R

VIR
→ D

→ #
→ B
→ d
→ L
→ D
→ c
→ a
→ 1
→ 0
→ 0
→ 0
→ 2
→ n
→ f
→ i
→

→ #
→ I
→ t
→ o
→ w
→ t
→ h
→ n
→ L
→ y
→ m
→ a
→ w
→ t

→ #
→ f
→
NET
→ G
→ 1
→ 0
→ 1

(continued from previous page)



FIX
→ R
→ 1
→ 0
→ 0
→ 2
FIX
→ N
→ S

→ #
→ L
→ a
→ o
→ t
→ f

LOG
→ \$
→ d
→ l

LOG
→ \$
→ l

IRO
→ V
→ L
→ D
→ \$
→ i
→ b
→ l

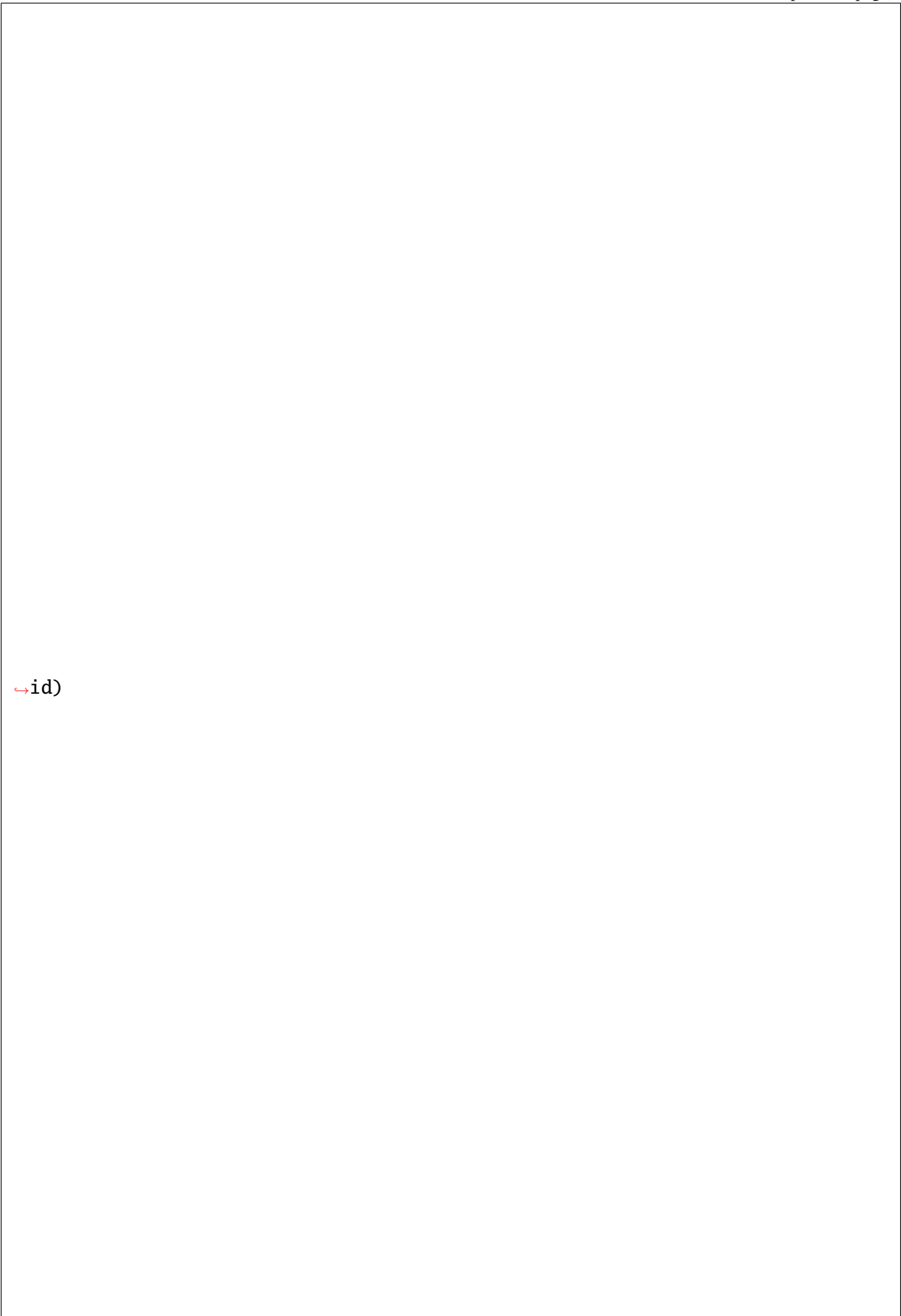
Af-
ter
the
en-
vi-
ron-
men-
is
built
you
can
cre-
ate
a
vol-
ume
with

request an instance with the volume to nova:



(continues on next page)

(continued from previous page)



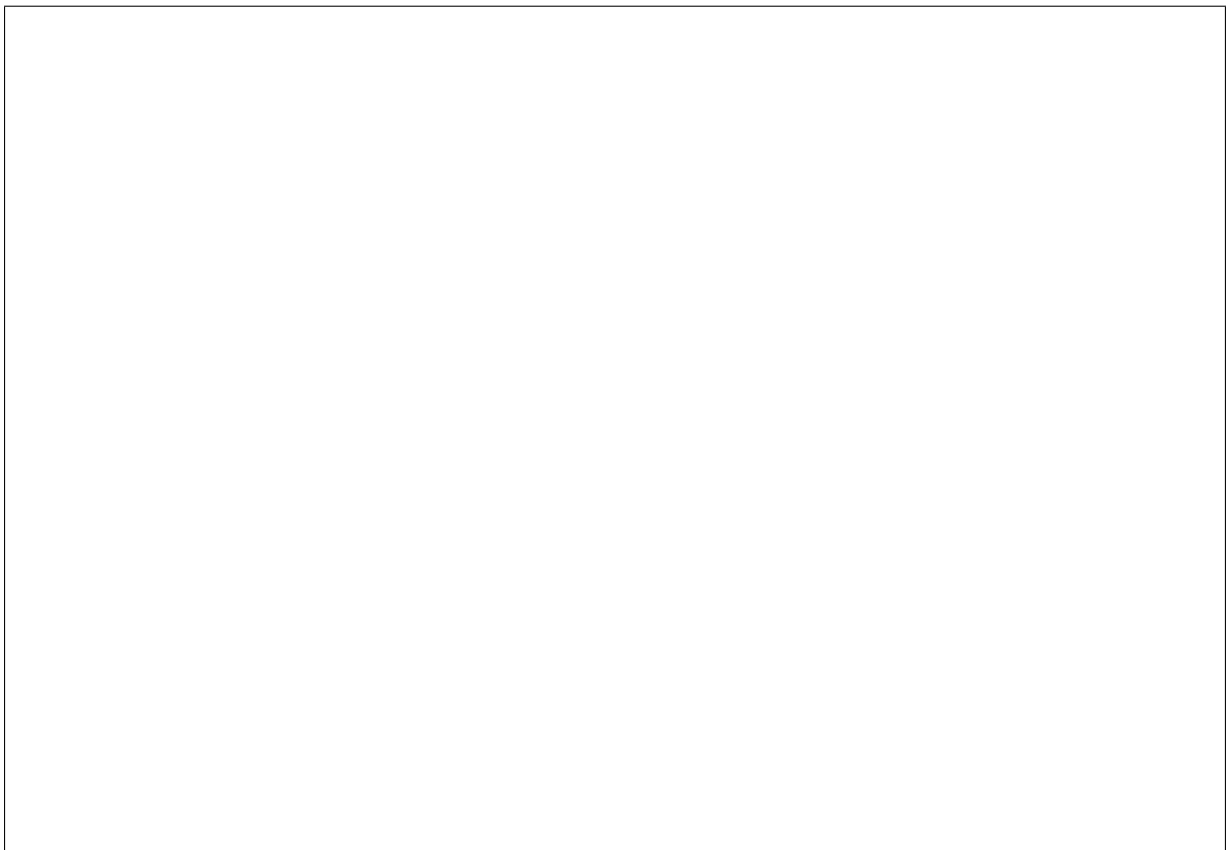
↪id)

↪#
↪C:
↪v
vol
↪\$
↪v
↪C:
↪-
↪-
↪i
↪
↪\$
↪-
↪-
↪S
↪1.
↪m
↪v
↪-
↪f
↪v
↪-
↪C

↪#
↪s
↪i
oper
↪S
↪C:
↪-
↪-
↪f
↪b
↪-
↪-
↪v
↪
↪\$
↪-
↪-
↪k
↪n
↪d
↪t

You
can
also
run
an
in-
te-
gra-
tion
test
that
an
in-
stanc
is
boot
from
a
re-
mote

volume with tempest in the environment:



cd
→/
→o
→S
→t
tox.
→-
→e
→a
→p
→-
→-
→
→i
→t
→p
→t
→s
→t
→b
→b
→f
→v

Plea
note
that
the
stor-

of the node and the configuration present. As such a node does not exclusively have to boot via a remote volume, and as such *validate* actions upon nodes may be slightly misleading. If an appropriate *volume target* is defined, no error should be returned for the boot interface.

ironic

ironic package

Subpackages

ironic.api package

Subpackages

ironic.api.controllers package

Subpackages

ironic.api.controllers.v1 package

Submodules

age
in-
ter-
face
will
only
in-
di-
cate
er-
rors
base
upon
the
state

8.1.
Iron
Ser
Pyt
API
Ref
er-
enc

ironic.api.controllers.v1.allocation module

class ir

Base
pec
res
Res
RES
con-
troll
for
al-
lo-
ca-
tion

delete()

Dele
an
al-
lo-
ca-
tion.

Parame

all
UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

get_all

Re-
triev
a
list
of
al-
lo-
ca-

tions

Parameters

- **node**
The name of the node to get only allocations for that node.
- **resource**
The resource class.
- **state**
The state of the allocation.
- **marker**
The marker for

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

large
data
sets.

- **limit**
maximum
i-
num
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

returned.

- **file**
Optionally, you can pass a list with a specific identifier set of fields of the resource to be

- **owner**
Filter by owner

get_one

Retrieves information about the given allocation.

Parameters

- **all**
UI or log-

returned.

i-
cal
nam
of
an
al-
lo-
ca-
tion.

- **file:**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid,

patch(al

Up-
date
an
ex-
ist-
ing
al-
lo-
ca-
tion.

Parame

- **all**
UU

or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
al-
lo-
ca-
tion.

post (*alle*)

Cre-
ate
a
new
al-
lo-
ca-
tion.

Parame

all
an
al-
lo-
ca-
tion
with
the
re-
ques
body

class ir

Base
pec
res
Res
RES
con-
troll
for
al-
lo-
ca-
tion

delete()

get_all

invalid

ironic.a

ironic.a

ironic.a

ironic.a

ironic.api.controllers.v1.bios module

class ir

Base
pec
res
Res
RES
con-
troll
for
bios

get_all

List
node
bios
set-
ting

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
bios
set-
ting

Param

set
Log
i-
cal
nam
of
the
set-
ting
to
re-
triev

ironic.a

ironic.a

Buil
a
dict
con-
tain-
ing
a
bios
set-
ting
valu

ironic.api.controllers.v1.chassis module

class ir

Base
pec
res
Res
RES
con-
troll
for
Cha
sis.

delete(

Dele
a
chas
sis.

Paramet

cha
UUI
of
a
chas
sis.

detail(

Re-
triev
a
list
of
chas
sis
with
de-
tail.

Paramet

- **mar**
pag-
i-
na-
tion
mar

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault

asc.

get_all

Re-
triev
a
list
of
chas
sis.

Paramete

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **sor**
col-
umn

returned.

to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **fields**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

get_one

Re-
triev
in-
for-
ma-
tion

about
the
give
chas
sis.

Parame

- **cha**
UI
of
a
chas
sis.

- **fiel**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

invalid

nodes =
Ex-
pose
node
as
a
sub-
elem
of
chas
sis

patch(*cha*

Up-
date
an
ex-
ist-
ing
chas
sis.

Parame

- **cha**
UUI
of
a
chas
sis.

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
chas
sis.

post(*cha*

Cre-
ate
a
new
chas
sis.

Parame

cha
a
chas
sis
with
the

re-
ques
body

ironic.a

ironic.a

ironic.api.controllers.v1.collection module

ironic.a

Re-
turn
a
link
to
the
next
sub-
set
of
the
col-
lec-
tion.

ironic.a

Re-
turn
the
col-
lec-
tion
has
more
item

ironic.a

Buil
a
col-
lec-

tion
dict
in-
clud
ing
the
next
link
for
pag-
ing
sup-
port

Paramet

- **item**
List
of
un-
san-
i-
tized
item
to
in-
clud
in
the
col-
lec-
tion
- **item**
Nam
of
dict
key
for
item
valu
- **lim**
Pag-
ing
limi
- **url**
Bas

URI
for
buil
ing
next
link

- **file**
Op-
tion
field
to
use
for
san-
i-
tize
func
tion

- **san**
Op-
tion
san-
i-
tize
func
tion
run
on
each
item
item
char
will
be
done
in-
plac

- **key**
Key
nam
for
buil
ing
next
URI

-

kwargs
othe
ar-
gu-
men
pass
to
get.

Returns

A
dict
con-
tain-
ing
iter
and
nex
val-
ues

ironic.api.controllers.v1.conductor module

class ir

Base
pec
res
Res
RES
con-
troll
for
con-
duc-
tors.

get_all

Re-
triev
a
list
of
con-
duc-
tors.

Parame

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **marl**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This
- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.
- **sor**
di-
rec-
tion

returned.

tail.

to
sort.
asc
or
desc
De-
fault
asc.

- **file:**
Optionally
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **det:**
Optionally
bool
to
in-
di-
cate
when
re-
triev
a
list
of
con-
duc-
tors
with
de-

get_one

Re-
triev-
in-
for-
ma-
tion
about
the
give
con-
duc-
tor.

Paramet

- **hos**
host
nam
of
a
con-
duc-
tor.
- **fie**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

invalid

ironic.a

ironic.a

ironic.api.controllers.v1.deploy_template module

class ir

Base
pec
res
Res
RES
con-
troll
for
de-
ploy
tem-
plate

delete(/

Dele
a
de-
ploy
tem-
plate

Parame

tem
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

get_all

Re-

triev
a
list
of
de-
ploy
tem-
plate

Parame

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **sor**
col-
umn
to
sort

returned.

re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.
- **file**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be
- **det**
Op-
tion
bool
to
in-
di-
cate
whe

detail.

re-
triev
a
list
of
de-
ploy
tem-
plate
with

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
de-
ploy
tem-
plate

Parame

- **temp**
UI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate
- **file**
Op-
tion:
a
list
with

returned.

a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid

patch(*te*

Up-
date
an
ex-
ist-
ing
de-
ploy
tem-
plate

Paramete

- **temp**
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate
- **pat**
a
json
PAT

doc-
u-
men
to
ap-
ply
to
this
de-
ploy
tem-
plate

post (*tem*

Cre-
ate
a
new
de-
ploy
tem-
plate

Parame

tem
a
de-
ploy
tem-
plate
with
the
re-
ques
body

ironic.a

ironic.a

Add
links
to
the
de-
ploy
tem-
plate

ironic.a

Ar-

gu-
men
val-
ida-
tor
to
chec
tem-
plate
for
du-
pli-
cate
step

ironic.a

ironic.a

ironic.a

Re-
mov
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Paramet
fie

(lis
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
then
all

ironic.api.controllers.v1.driver module

class ir

Base
pec
res
Res
RES
con-
troll
for
driv
pass
This
con-
troll
al-
low
ven-
dors
to
ex-
pose
cros
node
func
tion-
al-
ity
in
the
Iron

Ironic will merely relay the message from here to the specified driver, no introspection will be made in the message body.

API

methods

Re-
triev
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the
give
driv

Parameters

driver
name
of
the
driv

Returns

dic-
tio-
nary
with
<ver
dor
meth
name
meta
data
en-
tries

Raises

Driv
Not-
Four
if
the
driv
name
is
in-

valid
or
the
drive
can-
not
be
load

class ir

Base
pec
res
Res

logical

Re-
turn
the
log-
i-
cal
disk
prop
er-
ties
for
the
drive

Parame

drive
Nam
of
the
drive

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
prop
er-
ties
that
can

cal disks and a textual description for them.

be
men
tion
for
log-
i-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
driv
does
sup-
port
RAI
con-
fig-
u-
ra-
tion.

Raises

No-
tAc-
cept
able
if
re-
ques
ver-
sion
of
the
API
is
less
than
1.12

Raises

Driv
Not-
Four
if
driv

is
not
load
on
any
of
the
con-
duc-
tors.

class ir

Base
pec
res
Res
RES
con-
troll
for
Driv

get_all

Re-
triev
a
list
of
driv

get_one

Re-
triev
a
sin-
gle
driv

propert

Re-
triev
prop
erty
in-
for-
ma-
tion
of
the

give
drive

Parameters

drive
name
of
the
drive

Returns

dictionary
with
<property
name
description
entries

Raises

Drive
Not-
Found
(HTTP
404)
if
the
drive
name
is
in-
valid
or
the
drive
cannot
be
loaded

raid = <

Example
RAID
as
a
sub-
element

of
drive

vendor_p
<ironic

ironic.a

Con
vert
drive
type
info
to
a
dict.

Paramet

- **name**
name
of
a
hard
ware
type
- **hos**
list
of
con-
duc-
tor
host
name
drive
is
ac-
tive
on.
- **det.**
bool
whe
to
in-
clud
de-
taile
info.

faces fields.

such
as
the
type
field
and
de-
fault
in-
ter-

- **int**
op-
tion:
list
of
dicts
of
hard
ware
in-
ter-
face
info.

Returns
dict
rep-
re-
sent
ing
the
drive
ob-
ject.

ironic.a

This
meth
hide
field
that
were
adde
in
new
API
ver-
sion

Cer-
tain
field
were
in-
tro-
duce
at
cer-
tain
API
ver-
sion
The
field
are
only
mad
avai
able

when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.a

Con
vert
driv
and
hard
ware
type
to
an
API
seria
ob-
ject.

Paramet

- **har**
dict
map
ping
hard
ware
type
nam
to
con-
duc-

faces fields.

ironic.api.controllers.v1.event module

tor
host
nam

•
det.
bool
whe
to
in-
clud
de-
taile
info.
such
as
the
type
field
and
de-
fault
in-
ter-

Returns
an
API
serial
driv
col-
lec-
tion
ob-
ject.

class ir

Base
pec
res
Res

RES
con-
troll
for
Ever

ironic.api.controllers.v1.node module

post(*evt*,

ironic.a

Val-
ida-
tor
for
even

class ir

Base
pec
res
Res

get(*node*

Get
the
cur-
rent
boot
de-
vice
for
a
node

Paramete

node
the
UI
or
log-
i-
cal
nam
of
a
node

Returns

a
json
ob-
ject

unknown.

con-
tain-
ing:

boot_c
the
boot
de-
vice
one
of
ironic
comm
boo
or
Non
if
it
is
un-
know

persist
When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

put (*node*)
Set
the
boot
de-
vice
for
a

node
Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Paramete

- **node**
the
UI
or
log-
i-
cal
nam
of
a
node
- **boot**
the
boot
de-
vice
one
of
iron
com
boo
- **per:**
Boo
valu
True
if
the
boot
de-

not. Default: False.

vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

support

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

node
the
UI
or
log-
i-
cal
nam
of
a
node

Returns

A
json
ob-
ject
with
the
list
of
sup-
port
boot
de-
vice

class `ir`

Base
obj

class `ir`

Base
pec
res
Res

get_all

Get
node
hard
ware
com
po-
nent
and
their
in-
di-
ca-
tors.

Param

node
the
UUI
or
log-
i-
cal
nam
of
a
node

Returns

A
json
ob-
ject
of
hard
ware
com
po-
nent
(*irc*)

tor IDs (from *get_supported_indicators*) as values.

com
com
as
keys
with
in-
di-
ca-

get_one

Get
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
and
its
state

Paramete

- **node**
the
UI
or
log-
i-
cal
nam
of
a
node
- **ind**
In-
di-
ca-
tor
ID
(as
re-
port

by
get_

Returns

a
dict
with
the
state
key
and
one
of
mod
as
a
valu

put (*node*

Set
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
to
the
de-
sirec
state

Parame

- **node**
the
UI
or
log-
i-
cal
nam
of
a
node
-

ind.
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

- **sta**
In-
di-
ca-
tor
state
one
of
mod

class ir

Base
pec
res
Res

put(*node*

In-
ject
NM
for
a
node

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-

di-
ately

Parame

node

the

UI

or

log-

i-

cal

nam

of

a

node

Raises

Not-

Four

if

re-

ques

ver-

sion

of

the

API

does

sup-

port

in-

ject

nmi.

Raises

HTT

For-

bid-

den

if

the

pol-

icy

is

not

au-

tho-

rize

Raises

Nod

Not-

Four

if
the
node
is
not
found

Raises

Node
Locked
if
the
node
is
locked
by
another
conductor.

Raises

Unsupported
Driver
extension
if
the
node
driver
does
not
support
management
or
man

agement.inject_nmi.

Raises

Invalid
Parameter
Entered
Value
when

valid boot device is specified.

the
wron
driv
info
is
spec
i-
fied
or
an
in-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info.

class ir

Base
pec
res
Res

get(*node*)

Get
con-
nec-
tion
in-
for-
ma-
tion
abou
the
con-
sole

Parameter
node
URI

or
log-
i-
cal
nam
of
a
node

put (*node*)

Star
and
stop
the
node
con-
sole

Paramete

- **node**
UUI
or
log-
i-
cal
nam
of
a
node

- **enal**
Boo
valu
whe
to
en-
able
or
dis-
able
the
con-
sole

class ir

Bas
pec
res

Res

delete(*n*)

Re-
mov
the
node
from
main
te-
nan
mod

Parame

node
the
UI
or
log-
i-
cal
nam
of
a
node

put(*node*)

Put
the
node
in
main
te-
nan
mod

Parame

- **node**
the
UI
or
log-
i-
cal_
of
a
node
-

rea
Op-
tion
the
rea-
son
why
its
in
main
te-
nan

class ir

Base
pec
res
Res

boot_dev

Ex-
pose
boot
as
a
sub-
elem
of
man
age-
men

indicat

Ex-
pose
in-
di-
ca-
tors
as
a
sub-
elem
of
man
age-
men

inject_r

Ex-
pose

in-
ject_
as
a
sub-
elem
of
man
age-
men

class ir

Base
pec
res
Res

console

Ex-
pose
con-
sole
as
a
sub-
elem
of
state

get(*node*)

List
the
state
of
the
node

Param

node
the
UUID
or
log-
i-
cal_
of
a
node

power(*no*)

Set

the
pow
state
of
the
node

Parame

- **node**
the
UUU
or
log-
i-
cal
nam
of
a
node
- **tar**
The
de-
sired
pow
state
of
the
node
- **time**
time
out
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state

indicates to use default timeout.

is in CLEANING state.

Non

Raises

Client
error
(HTTP
409)
if
a
power
operation
action
is
already
read
in
program

Raises

Invalid
State
(HTTP
400)
if
the
request
target
state
is
not
valid
or
if
the
node

Raises

Not
Acceptable
(HTTP
406)
for
software
re-

ter, if requested version of the API is less than 1.27.

boot
soft
pow
off
or
time
out
pa-
ram-
e-

Raises

In-
valid
(HT
400)
if
time
out
valu
is
less
than
1.

provisio

Asy
chro
trig-
ger
the
pro-
vi-
sion
ing
of
the
node

This
will
set
the
tar-
get
pro-
vi-
sion
state
of
the

gin which actually applies the state change. This call will return a 202 (Accepted) indicating the request was accepted and is in progress; the client should continue to GET the status of this node to observe the status of the requested action.

Parameters

node
and
a
back
group
task
will
be-

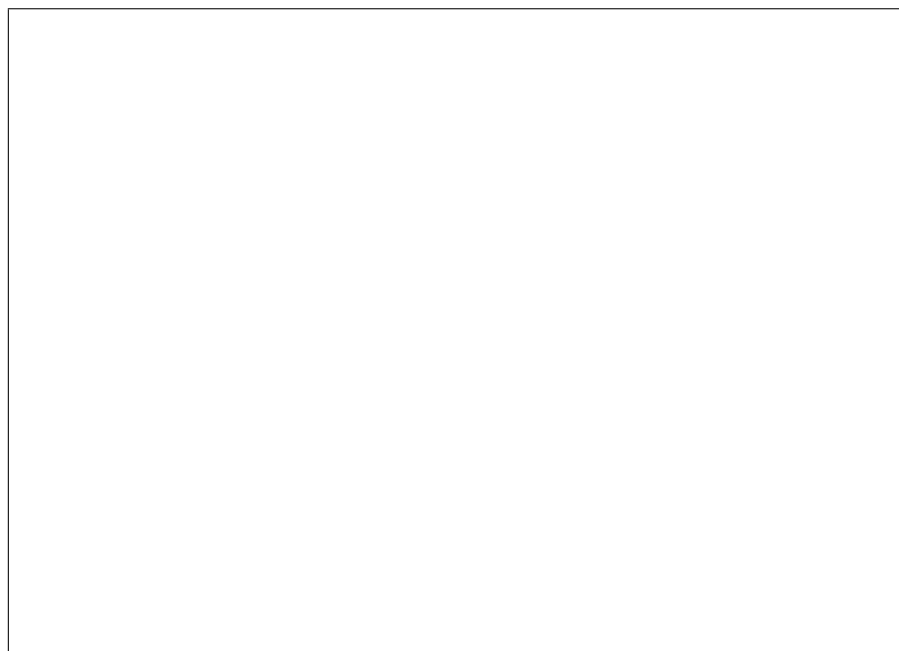
- **node**
UUID
or
log-
i-
cal
name
of
a
node

- **target**
The
de-
sired
pro-
vi-
sion
state
of
the
node
or
verb

- **compression**
Optional
A
gzip
and
base
en-
code
con-

figdrive from. Only valid when setting provision state to active or rebuild.

ing step is a dictionary with required keys interface and step, and optional key args. If specified, the value for args is a keyword variable argument dictionary that is passed to the cleaning step method.:

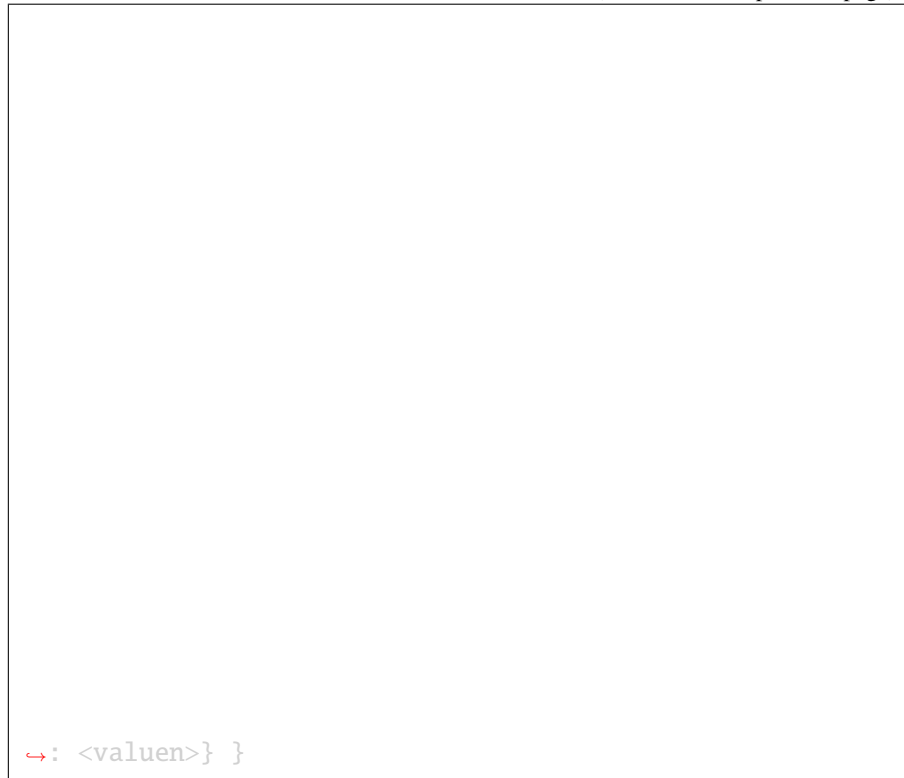


(continues on next page)

fig-
driv
or
a
dict
to
buil
a
con-

•
cle
An
or-
dere
list
of
clea
ing
step
that
will
be
per-
form
on
the
node
A
clea

(continued from previous page)

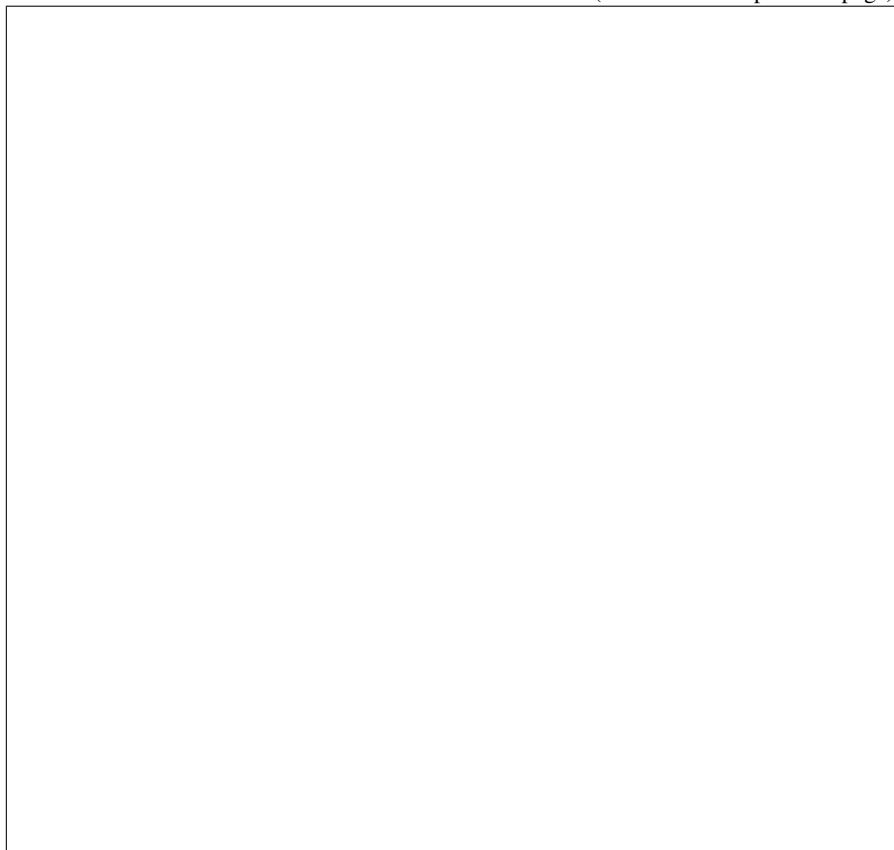


For
ex-
am-
ple
(this
isnt
a
real
ex-
am-
ple,
this
clear
ing
step
does
ex-
ist):



(continues on next page)

(continued from previous page)

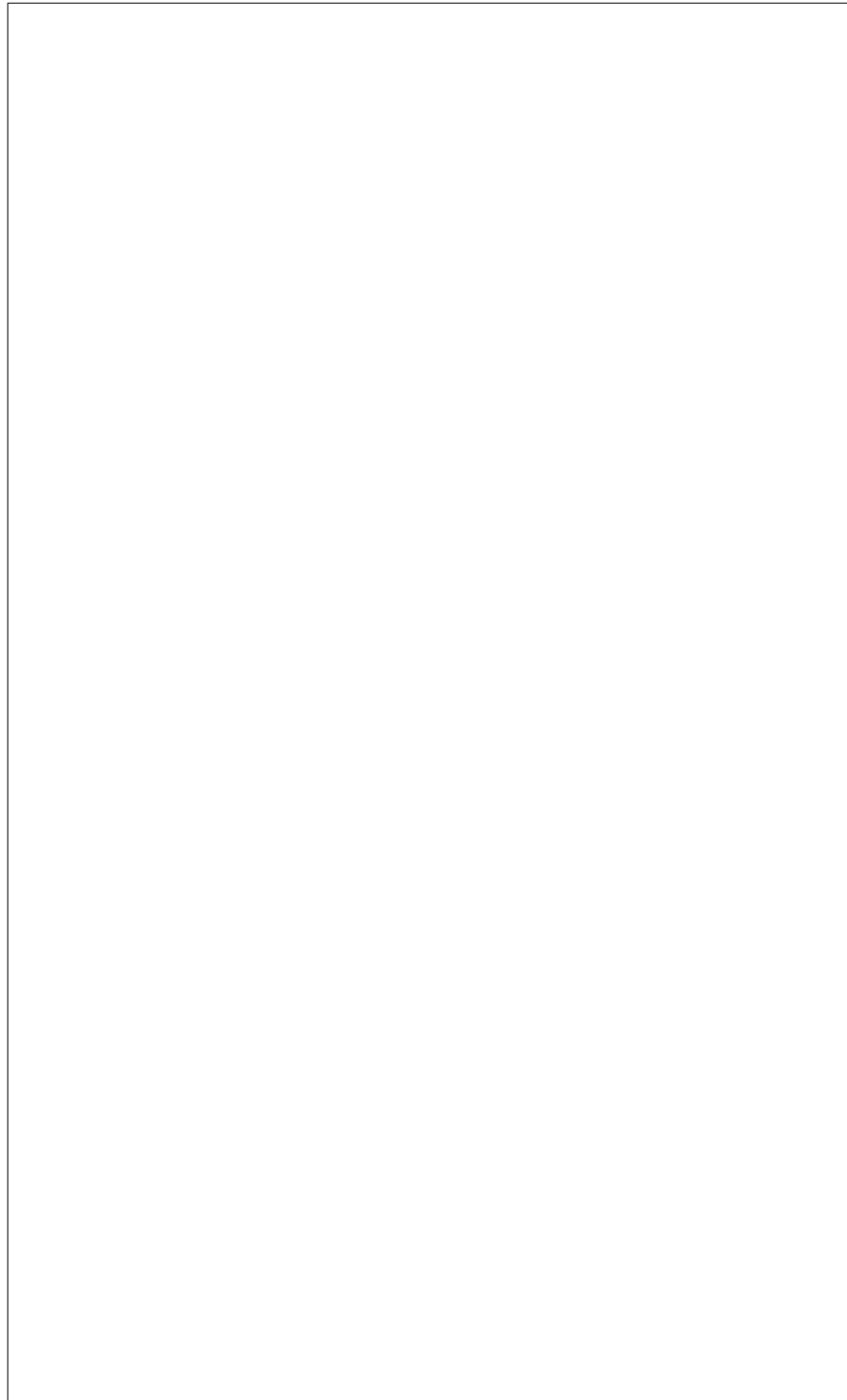


This
is
re-
quir
(and
only
valid
whe
tar-
get
is
clea

- **dep**
A
list
of
de-
ploy
steps
that
will
be
per-
form

on
the
node
A
de-
ploy
step

is a dictionary with required keys interface, step, priority and args. If specified, the value for args is a keyword variable argument dictionary that is passed to the deploy step method.:



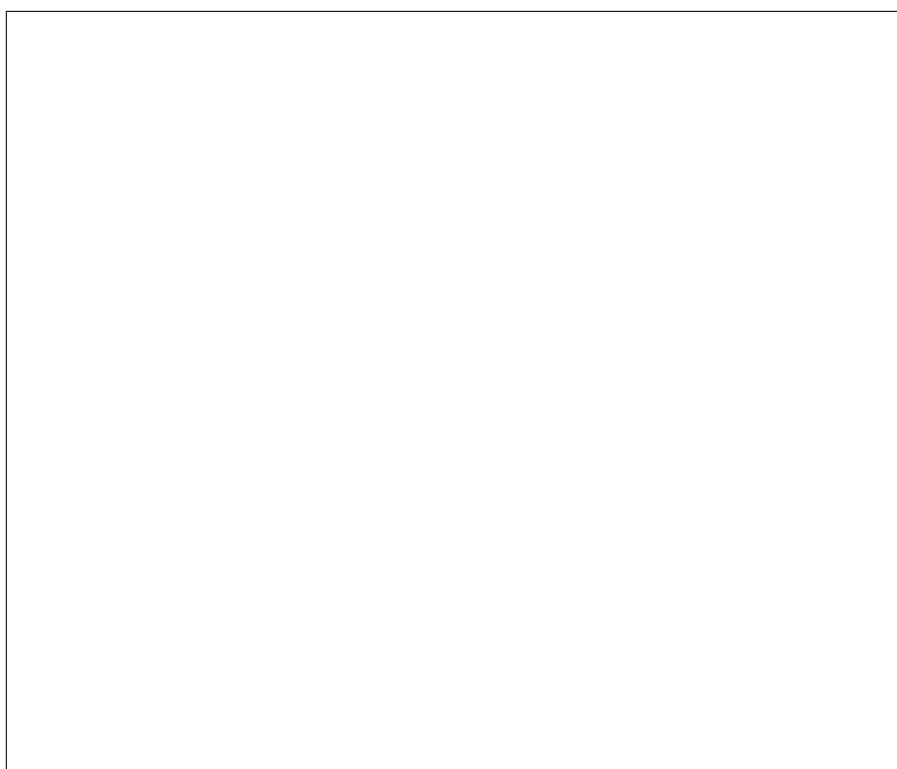
(continues on next page)

↔: <valuen>

(continued from previous page)

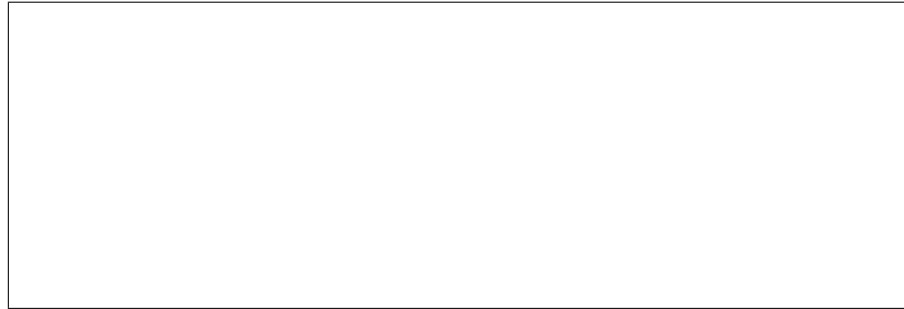


For
ex-
am-
ple
(this
isnt
a
real
ex-
am-
ple,
this
de-
ploy
step
does
ex-
ist):



(continues on next page)

(continued from previous page)



vironment. This is required (and only valid), when target is rescue.

This
is
used
only
when
target
is
active
or
re-
build
and
is
optiona

- **rescue**
A
string
represent
ing
the
password
to
be
set
inside
the
rescue
en-

- **dis**

When
to
skip
boot
ing
rame
for
clear
ing.

Raises

Node
Lock
(HTTP
409)
if
the
node
is
currentl
lock

Raises

Client
ror
(HTTP
409)
if
the
node
is
al-
read
be-
ing
pro-
vi-
sion

Raises

In-
valid
Pa-
ram-
e-
ter-
Value
(HTTP
400)
if
val-
i-

power driver interface fails.

current state.

node is in maintenance mode.

da-
tion
of
clea
de-
ploy
or

Raises

In-
valic
State
(HT
400)
if
the
re-
ques
tran-
si-
tion
is
not
pos-
si-
ble
from
the

Raises

Nod
Mai
te-
nanc
(HT
400)
if
op-
er-
a-
tion
can-
not
be
per-
form
be-
caus
the

requested state transition or parameters.

Raises

NoF
duc-
tor-
Wor
(HT
503),
if
no
worl
ers
are
avai
able

Raises

No-
tAc-
cept
able
(HT
406),
if
the
API
ver-
sion
spec
i-
fied
does
not
al-
low
the

raid(*noo*

Set
the
tar-
get
raid
con-
fig
of
the
node

Parame

dictionary as well.

- **node**
the
UUI
or
log-
i-
cal
nam
of
a
node
- **target**
De-
sirec
tar-
get
RAI
con-
fig-
u-
ra-
tion
of
the
node
It
may
be
an
emp

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
RAI
con-

tion.

fig-
u-
ra-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
tar-
get
raid
con-
fig
fails

Raises

No-
tAc-
cept
able
if
re-
ques
ver-
sion
of
the
API
is
less
than
1.12

class ir

Base
pec
res
Res

delete()

Re-
mov
one
or
all
trait
from
a
node

Parame

tra
Strin
valu
trait
to
re-
mov
from
a
node
or
Non
If
Non
all
trait
are
re-
mov

get_all

List
node
trait

put(*traits*

Add
a
trait
to
a
node

Parame

- **tra**
Strin
valu
trait

If not None, adds this trait to the node.

trait. If not None, replaces the nodes traits with this list.

to
add
to
a
node
or
Non
Mu-
tu-
ally
ex-
clu-
sive
with
trait

- **tra**
List
of
Strin
trait
to
set
for
a
node
or
Non
Mu-
tu-
ally
ex-
clu-
sive
with

class ir

Base
pec
res
Res

delete(

De-
tach
a
VIF

from
this
node

Parameter

vif

The
ID
of
a
VIF
to
de-
tach

get_all

Get
a
list
of
at-
tach
VIF

post(vif

At-
tach
a
VIF
to
this
node

Parameter

vif

a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.
It
must
have
an
id

whose value is a unique identifier for that VIF.

API. Ironic will merely relay the message from here to the appropriate driver, no introspection will be made in the message body.

key,

class ir

Base
pec
res
Res
RES
con-
troll
for
Ven-
dor-
Pass

This
con-
troll
al-
low
ven-
dors
to
ex-
pose
a
cus-
tom
func
tion-
al-
ity
in
the
Iron

methods

Re-
triev
in-
for-
ma-
tion
abou
ven-
dor
meth

ods
of
the
give
node

Parameters

node
UUID
or
log-
i-
cal
nam
of
a
node

Returns

dic-
tio-
nary
with
<ver
dor
meth
nam
meta
data
en-
tries

Raises

Nod
Not-
Four
if
the
node
is
not
foun

class ir

Base
pec
res
Res

RES
con-
troll
for

Nod

delete()

Dele
a
node

Parame

nod
UUI
or
log-
i-
cal
nam
of
a
node

detail()

n
s
c
l
Re-
triev
a
list
of
node
with
de-
tail.

Parame

- **cha**
Op-
tion
UUI
of
a
chas
sis,
to
get
only
node
for
that

stance.

ated nodes. May be combined with other parameters.

chas
sis.

- **ins**
Op-
tion:
UUI
of
an
in-
stan-
to
find
the
node
as-
so-
ci-
ated
with
that
in-

- **ass**
Op-
tion:
bool
whe
to
re-
turn
a
list
of
as-
so-
ci-
ated
or
unas
so-
ci-

- **mai**
Op-
tion:
bool
valu

or not in maintenance mode (False).

that
in-
di-
cate
whe
to
get
node
in
main
te-
nan
mod
(Tru

- **ret.**
Op-
tion:
bool
valu
that
in-
di-
cate
whe
to
get
node
whic
are
re-
tired

- **pro**
Op-
tion:
strin
valu
to
get
only
node
in
that
pro-
vi-
sion
state

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

marl
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**
di-
rec-
tion
to

sort.
asc
or
desc
De-
fault
asc.

- **drive**
Option:
string
value
to
get
only
node
us-
ing
that
drive

- **resource**
Option:
string
value
to
get
only
node
with
that
re-
sour

- **fault**
Option:
string
value
to
get
only
node
with
that
fault

-

con
Op-
tion:
strin
valu
to
get
only
node
with
that
con-
duc-
tor_

- **own**
Op-
tion:
strin
valu
that
set
the
own
who
node
are
to
be
retru

- **les**
Op-
tion:
strin
valu
that
set
the
lesse
who
node
are
to
be
re-
turn

- **pro**
Op-

to be returned.

tion
strin
valu
that
set
the
proj
-
less
or
own
-
who
node
are

- **des**
Op-
tion
strin
valu
to
get
only
node
with
de-
scrip
tion
field
con-
tains
mat
ing
valu

from_cha

A
flag
to
in-
di-
cate
if
the
re-
ques
to

top-level resource Chassis

this
con-
troll
are
com
ing
from
the

get_all

Re-
triev
a
list
of
node

Paramete

- **cha**
Op-
tion:
UUI
of
a
chas
sis,
to
get
only
node
for
that
chas
sis.

- **ins**
Op-
tion:
UUI
of
an
in-
stan
to

stance.

ated nodes. May be combined with other parameters.

find
the
node
as-
so-
ci-
ated
with
that
in-

- **ass**
Op-
tion:
bool
whe
to
re-
turn
a
list
of
as-
so-
ci-
ated
or
unas
so-
ci-

- **mai**
Op-
tion:
bool
valu
that
in-
di-
cate
whe
to
get
node
in
mai
te-
nan

or not in maintenance mode (False).

mod
(Tru

- **ret.**
Op-
tion:
bool
valu
that
in-
di-
cate
whe
to
get
re-
tired
node

- **pro**
Op-
tion:
strin
valu
to
get
only
node
in
that
pro-
vi-
sion
state

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **lim**
max

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **drive**
Op-
tion:
string
valu

to
get
only
node
us-
ing
that
drive

- **res**
Op-
tion
strin
valu
to
get
only
node
with
that
re-
sour

- **con**
Op-
tion
strin
valu
to
get
only
node
with
that
con-
duc-
tor_

- **con**
Op-
tion
strin
valu
to
get
only
node
man
aged
by

that
con-
duc-
tor.

- **own**
Op-
tiona
strin
valu
that
set
the
own
who
node
are
to
be
retru

- **les**
Op-
tiona
strin
valu
that
set
the
lesse
who
node
are
to
be
re-
turn

- **pro**
Op-
tiona
strin
valu
that
set
the
proj
-

lesse

to be returned.

returned.

or
own
-

who
node
are

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **fault**
Op-
tion:
strin
valu
to
get
only
node
with
that
fault

- **des**
Op-
tion:
strin
valu
to
get

only
node
with
de-
scrip-
tion
field
con-
tains
matc
ing
valu

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
node

Parame

- **node**
UUID
or
log-
i-
cal
nam
of
a
node
- **file**
Op-
tiona
a
list
with
a
spec
i-
fied
set

returned.

of
field
of
the
re-
sour
to
be

invalid
'instanc
'raid_co

maintena
object>

Ex-
pose
main
te-
nanc
as
a
sub-
elem
of
node

managem
object>

Ex-
pose
man
age-
men
as
a
sub-
elem
of
node

patch(*no*

Up-
date
an
ex-
ist-
ing
node

Parame

ing the driver field.

- **node**
UI
or
log-
i-
cal
nam
of
a
node

- **res**
whe
to
re-
set
hard
ware
in-
ter-
face
to
their
de-
fault
Only
valid
whe
up-
dat-

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
node

post(*no*

Cre-
ate

a
new
node

Parameter

node
a
node
with
the
re-
ques
body

states =

Ex-
pose
the
state
con-
troll
ac-
tion
as
a
sub-
elem
of
node

validate

Val-
i-
date
the
drive
in-
ter-
face
us-
ing
the
node
UUI
or
nam

Note
that
the
node
in-

ter-
face
is
dep-
re-
cate
in
favo
of
the
node
in-
ter-
face

Parame

- **node**
UI
or
nam
of
a
node
- **node**
UI
of
a
node

vendor_p <ironic

A
re-
sour
used
for
ven-
dors
to
ex-
pose
a
cus-
tom
func
tion-
al-
ity

API

in
the

ironic.a

ironic.a

This
meth
hide
field
that
were
adde
in
new
API
ver-
sion

Cer-
tain
node
field
were
in-
tro-
duce
at
cer-
tain
API
ver-
sion
The
field
are
only
mad
avai

able when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.a

Add
links
to
the
in-
di-

ca-
tor.
ironic.a
ironic.a
ironic.a
ironic.a
ironic.a
ironic.a
ironic.a

Re-
mov
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Paramet
fie:
(*lis*
of
str
list
of

field
to
pre-
serv
or
Non
to
pre-
serv
them
all

ironic.a

ironic.a

ironic.a

ironic.a

When
cre-
at-
ing
an
ob-
ject,
re-
ject
field
that
ap-
pear
in
new
ver-
sion

ironic.a

ironic.a

Cha
pro-
vi-
sion
state
nam
for
API
back
ward

method.

com
pat-
i-
bil-
ity.

Parameter

obj

The
dict
be-
ing
re-
turn
to
the
API
client
that
is
to
be
up-
date
by
this

`ironic.a`

Val-
i-
date
node
net-
work
field

This
meth
val-
i-
date
net-
work
data
con-
fig-
u-
ra-
tion
agai
JSO

sche

Paramet

net

a

net-

worl

field

to

val-

i-

date

Raises

In-

valid

if

net-

worl

data

is

not

sche

com

ironic.api.controllers.v1.notification_utils module

ironic.a

Help

for

emit

ting

API

end

no-

ti-

fi-

ca-

tions

Paramet

•

con

re-

ques

con-

text.

•

obj
re-
sour
rpc
ob-
ject.

- **act:**
Ac-
tion
strin
to
go
in
the
Ever
Type

- **kwa:**
kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-
tion
pay-
load

ironic.a

Help
for
emit
ting
API
start
no-
ti-
fi-
ca-
tions

Paramet

- **con**
re-
ques
con-
text.

- **obj**
re-
sour
rpc
ob-
ject.

- **act:**
Ac-
tion
strin
to
go
in
the
Ever
Type

- **kwa:**
kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-
tion
pay-
load

ironic.a

Con
text
man

ager
to
han-
dle
any
er-
ror
no-
ti-
fi-
ca-
tions

Parameter

- **con**
re-
ques-
con-
text.
- **obj**
re-
sour-
rpc
ob-
ject.
- **act.**
Ac-
tion
strin
to
go
in
the
Ever
Type
- **kwa**
kwa
to
use
whe
cre-
at-
ing
the
no-

ironic.api.controllers.v1.port module

ti-
fi-
ca-
tion
pay-
load

class ir

Base
pec
res
Res
RES
con-
troll
for
Port

advanced

delete(*url*)

Dele
a
port

Parameter

por
UUI
of
a
port

Raises

Op-
er-
a-
tion
Per-
mit-
ted,
HTT
Not-
Four

detail(*url*)

S
Re-

triev
a
list
of
port
with
de-
tail.

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of
the
node
in-
ter-
face

Parame

- **node**
UI
or
nam
of
a
node
to
get
only
port
for
that
node
- **node**
UI
of
a

node
to
get
only
port
for
that
node

- **add:**
MA
ad-
dres
of
a
port
to
get
the
port
which
has
this
MA
ad-
dres

- **por:**
UI
or
nam
of
a
port
grou
to
get
only
port
for
that
port
grou

- **mar:**
pag-
i-
na-
tion
mar

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault

asc.

Raises

No-
tAc-
cept
able
HTT
Not-
Four

get_all

Re-
triev
a
list
of
port

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of
the
node
in-
ter-
face

Paramete

- **node**
UUU
or
nam
of
a
node
to
get
only

port
for
that
node

- **node**
UI
of
a
node
to
get
only
port
for
that
node

- **add**
MA
ad-
dres
of
a
port
to
get
the
port
whic
has
this
MA
ad-
dres

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **lim**
max

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file**
Op-
tion:
a
list

returned.

with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **por**
UI
or
nam
of
a
port
grou
to
get
only
port
for
that
port
grou

Raises
No-
tAc-
cept
able
HTT
Not-
Four

get_one

Re-
triev
in-
for-
ma-
tion
about
the

give
port

Parameters

- **port**
URI
of
a
port
- **fields**
Optional
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

Raises

No-
tAc-
cept
able
HTT
Not-
Four

invalid_
'local_'

patch(*pa*

Up-
date
an
ex-
ist-

ing
port

Parame

- **por-**
UUI
of
a
port

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
port

Raises

No-
tAc-
cept
able
HTT
Not-
Four

post (*por*

Cre-
ate
a
new
port

Parame

por-
a
port
with
the
re-
ques
body

Raises

No-
tAc-
cept
able
HTT
Not-
Four
Con
flict

ironic.a

ironic.a

ironic.a

ironic.a

Re-
mov
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Paramet

file
(lis
of
str
list

of
field
to
pre-
serv
or
Non
to
pre-
serv
then
all

ironic.api.controllers.v1.portgroup module

class ir

Base
pec
res
Res
RES
con-
troll
for
port
grou

delete()

Dele
a
port
grou

Parame

por
UUI
or
log-
i-
cal
nam
of
a
port
grou

detail()

Re-

triev
a
list
of
port
grou
with
de-
tail.

Parame

- **node**
UI
or
nam
of
a
node
to
get
only
port
grou
for
that
node

- **add**
MA
ad-
dres
of
a
port
grou
to
get
the
port
grou
whic
has
this
MA
ad-
dres

- **mar**

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**
di-
rec-
tion
to
sort.

asc
or
desc
De-
fault
asc.

get_all

Re-
triev
a
list
of
port
grou

Parame

- **node**
UI
or
nam
of
a
node
to
get
only
port
grou
for
that
node
- **add**
MA
ad-
dres
of
a
port
grou
to
get
the
port
grou
whic
has

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

this
MA
ad-
dres

- **marl**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

returned.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
port
grou

Parame

returned.

- **port**
UI
or
log-
i-
cal
nam
of
a
port
grou
- **file**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid

patch(*pa*

Up-
date
an
ex-
ist-
ing
port
grou

Parame

- **port**

UUI
or
log-
i-
cal
nam
of
a
port
grou

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
port
grou

post(*por*

Cre-
ate
a
new
port
grou

Parame

por
a
port
grou
with
the
re-
ques
body

ironic.a

Add
links
to
the

ironic.api.controllers.v1.ramdisk module

port
grou
ironic.a

class ir

Base
pec
res
Res
Con
troll
han-
dling
hear
beat
from
de-
ploy
ram

post(*no*

age
Pro-
cess
a
hear
beat
from
the
de-
ploy
ram

Paramete

- **node**
the
UUI
or
log-
i-
cal
nam

of
a
node

- **cal**
the
URI
to
reac
back
to
the
ram

- **age**
The
ver-
sion
of
the
age
that
is
hear
beat
ing.
Non
in-
di-
cate
that
the
age

that is heartbeating is a version before sending agent_version was introduced so agent v3.0.0 (the last release before sending agent_version was introduced) will be assumed.

- **age**
ran-
dom
gen-
er-
ated
val-
i-
da-
tion
to-
ken.

tus back to Ironic. The valid states are start, end, error

age
TLS
cer-
tifi-
cate
to
use
to
con-
nect
to
the
agen

- **age**
Cur-
rent
sta-
tus
of
the
hear
beat
ing
agen
Use
by
ana-
conc
rame
to
send
sta-

- **age**
Op-
tiona
sta-
tus
mes-
sage
de-
scrib
ing
cur-
rent
agen

Raises
Nod

Not-
Four
if
node
with
pro-
vide
UUI
or
nam
was
not
foun

Raises

In-
valic
uidC
Nam
if
node
is
not
valic
nam
or
UUI

Raises

No-
Vali
Hos
if
RPC
topic
for
node
coul
not
be
re-
triev

Raises

Not-
Four
if
re-
ques
API
ver-
sion
does

not
al-
low
this
end-
poin

class ir

Base
pec
res
Res
Con
troll
han-
dling
node
look
for
a
de-
ploy
rame

get_all

Loos
up
a
node
by
its
MA
ad-
dres
and
op-
tion-
ally
UU
If
the
re-
stric
op-
tion
is
set
to
True

tain transient states (e.g. deploy wait).

(the
de-
fault
limi
the
sear
to
node
in
cer-

Parameters

- **add**
list
of
MA
ad-
dres
for
a
node
- **node**
UUI
of
a
node

Raises

Not-
Four
if
re-
ques
API
ver-
sion
does
not
al-
low
this
end-
poin

Raises

Not-
Four
if

for the lookup.

suit-
able
node
was
not
foun
or
node
pro-
vi-
sion
state
is
not
al-
lowe

Raises

In-
com
plete
if
nei-
ther
node
UUI
nor
any
valid
MA
ad-
dres
was
pro-
vide

property

ironic.a

ironic.a

ironic.api.controllers.v1.utils module

class ir

Base
obj
Ob-
ject
to
hold
the
re-
spon-
sion
from
a
pass
call

obj

Stor
the
re-
sult
ob-
ject
from
the
view

status_c

Stor
an
op-
tion
sta-
tus_

ironic.a

Che
if
ager
to-
ken
is
avai
able

ironic.a

Che
if

tion to Ironic on heartbeat.

ager
ver-
sion
is
al-
lowe
to
be
pass
into
hear
beat

Ver-
sion
1.36
of
the
API
adde
the
abil-
ity
for
ager
to
pass
their
ver-
sion
in-
for-
ma-

`ironic.a`

Che
if
back
fill-
ing
al-
lo-
ca-
tions
is
al-
lowe

Ver-
sion
1.58

of
the
API
adde
sup-
port
for
back
fill-
ing
al-
lo-
ca-
tion

ironic.a

Che
if
al-
lo-
ca-
tion
own
field
is
al-
lowe

Ver-
sion
1.60
of
the
API
adde
the
own
field
to
the
al-
lo-
ca-
tion
ob-
ject.

ironic.a

Che
if
up-

dat-
ing
an
ex-
ist-
ing
al-
lo-
ca-
tion
is
al-
lowe
or
not.
Ver-
sion
1.57
of
the
API
adde
sup-
port
for
up-
dat-
ing
an
al-
lo-
ca-
tion.

`ironic.a`

Che
if
ac-
cess
ing
al-
lo-
ca-
tion
end-
point
is
al-
lowe
Ver-

for the node.

sion
1.52
of
the
API
ex-
pose
al-
lo-
ca-
tion
end-
poin
and
al-
lo-
ca-
tion,
field

ironic.a

Che
if
we
shou
sup-
port
bios
in-
ter-
face
and
end-
poin

Ver-
sion
1.40
of
the
API
add
sup-
port
for
bios
in-
ter-
face

ironic.a

Che
if
buil
ing
con-
fig-
driv
is
al-
lowe

Ver-
sion
1.56
of
the
API
adde
sup-
port
for
buil
ing
con-
fig-
driv

`ironic.a`

Che
if
con-
fig-
driv
can
con-
tain
a
ven-
dor_
key.

Ver-
sion
1.59
of
the
API
adde
sup-
port
for

con-
fig-
driv
ven-
dor_

ironic.a

Che
if
de-
ploy
are
avai
able

ironic.a

Che
if
ac-
cess
ing
de-
ploy
tem-
plate
end-
point
is
al-
lowe

Ver-
sion
1.55
of
the
API
ex-
pose
de-
ploy
tem-
plate
end-
point

ironic.a

Che
if
pass
ing

all the fields.

a
de-
tail=
quer
strin
is
al-
lowe

Ver-
sion
1.43
al-
lows
a
user
to
pass
the
de-
tail
quer
strin
to
list
the
re-
sour
with

ironic.a

Che
if
dy-
nam
driv
API
calls
are
al-
lowe

Ver-
sion
1.30
of
the
API
adde
sup-
port

calls in the /v1/drivers API.

the node object.

for
all
of
the
driv
com
po-
si-
tion
re-
latec

ironic.a

Che
if
dy-
nam
in-
ter-
face
field
are
al-
lowe

Ver-
sion
1.31
of
the
API
add
sup-
port
for
view
ing
and
set-
ting
the
field
in
V31.
on

ironic.a

Che
if

node.

ac-
cess
ing
con-
duc-
tor
end-
point
is
al-
lowe

Ver-
sion
1.49
of
the
API
ex-
pose
con-
duc-
tor
end-
point
and
con-
duc-
tor
field
for
the

ironic.a

Che
if
ac-
cess
ing
even
end-
point
is
al-
lowe

Ver-
sion
1.54
of
the

API
add
the
even
end-
point

ironic.a

Che
if
a
field
is
al-
lowe
in
the
cur-
rent
ver-
sion

ironic.a

Che
if
In-
ject
NM
is
al-
lowe
for
the
node

Ver-
sion
1.29
of
the
API
al-
lows
In-
ject
NM
for
the
node

ironic.a

Che
if
in-
spec
tion
abon
is
al-
lowe
Ver-
sion
1.41
of
the
API
add
sup-
port
for
in-
spec
tion
abon

`ironic.a`

Che
if
in-
spec
wait
is
al-
lowe
for
the
node
Ver-
sion
1.39
of
the
API
adds
in-
spec
wait
state
to
sub-

ing asynchronous hardware inspection.

ties.

sti-
tute
in-
spec
ing
state
dur-

ironic.a

Che
if
link
are
dis-
play

Ver-
sion
1.14
of
the
API
al-
lows
the
dis-
play
of
link
to
node
state
and
drive
prop
er-

ironic.a

Che
if
net-
worl
is
al-
lowe
in
port
link

ironic.a

ironic.a

Che
if
we
shou
sup-
port
node
re-
buil
with
con-
fig-
driv

Ver-
sion
1.35
of
the
API
adde
sup-
port
for
node
re-
buil
with
con-
fig-
driv

ironic.a

Che
if
we
shou
re-
turn
lo-
cal_
and
pxe_
field

Ver-
sion
1.19
of

the
API
adde
sup-
port
for
thes
new
field
in
port
ob-
ject.

`ironic.a`

Che
if
ac-
cess
ing
in-
ter-
nal_
is
al-
lowe
for
the
port

Ver-
sion
1.18
of
the
API
ex-
pose
in-
ter-
nal_
read
only
field
for
the
port

`ironic.a`

Che
if

port
is_s
field
is
al-
lowe

Ver-
sion
1.53
of
the
API
adde
is_s
field
to
the
port
ob-
ject.

ironic.a

Che
if
port
phys
i-
cal
net-
worl
field
is
al-
lowe

Ver-
sion
1.34
of
the
API
adde
the
phys
i-
cal
net-
worl
field
to
the

also check whether the target version of the Port object supports the `physical_network` field as this may not be the case during a rolling upgrade.

port
ob-
ject.
We

ironic.a

Che
if
mod
and
prop
er-
ties
can
be
adde
to/qu
from
a
port
grou

Ver-
sion
1.26
of
the
API
adde
mod
and
prop
er-
ties
field
to
port
grou
ob-
ject.

ironic.a

Che
if
we
shou
sup-
port
port

group
op-
er-
a-
tions
Ver-
sion
1.23
of
the
API
add-
sup-
port
for
Port
Gro

`ironic.a`

Che
if
port
group
can
be
used
as
sub-
con-
troll

Ver-
sion
1.24
of
the
API
add-
sup-
port
for
Port
group
as
sub-
con-
troll

`ironic.a`

Che
if

BIO
quer
shou
be
al-
lowe
base
on
ver-
sion

ironic.a

Che
if
RAI
con-
fig-
u-
ra-
tion
is
al-
lowe
for
the
node

Ver-
sion
1.12
of
the
API
al-
lows
RAI
con-
fig-
u-
ra-
tion
for
the
node

ironic.a

Che
if
hear
beat
and

look
end-
poin
are
al-
lowe

Ver-
sion
1.22
of
the
API
in-
tro-
duce
them

ironic.a

Che
if
chas
sis_
can
be
re-
mov
from
node

Ver-
sion
1.25
of
the
API
adde
sup-
port
for
chas
sis_
re-
mov

ironic.a

Che
if
we
shou
sup-
port

res-
cue
and
un-
res-
cue
op-
er-
a-
tions
and
in-
ter-
face

Ver-
sion
1.38
of
the
API
add
sup-
port
for
res-
cue
and
un-
res-
cue.

ironic.a

Che
if
pass
ing
a
re-
set_
quer
strin
is
al-
lowe

ironic.a

Che
if
Soft
Pow

Off
is
al-
lowe
for
the
node

Ver-
sion
1.27
of
the
API
al-
lows
Soft
Pow
Off,
in-
clud
ing
Soft
Re-
boot
for
the
node

ironic.a

Che
if
hear
beat
ac-
cept
ager
and
ager

ironic.a

Che
if
we
shou
sup-
port
stor-
age_
node
and

drive
field
Ver-
sion
1.33
of
the
API
add
sup-
port
for
stor-
age
in-
ter-
face

`ironic.a`

Che
if
trait
are
al-
lowe
for
the
node

Ver-
sion
1.37
of
the
API
al-
lows
trait
for
the
node

`ironic.a`

Che
if
hear
beat
ac-
cept
ager

ironic.a

Che
if
node
can
be
used

Ver-
sion
1.28
of
the
API
add
sup-
port
for
VIF
to
be
at-
tach
to
Nod

ironic.a

Che
if
vol-
ume
con-
nec-
tors
and
tar-
gets
are
al-
lowe

Ver-
sion
1.32
of
the
API
add
sup-
port
for

vol-
ume
con-
nec-
tors
and
tar-
gets

ironic.a

Ap-
ply
a
JSO
patc
one
op-
er-
a-
tion
at
a
time

If
the
patc
fails
to
ap-
ply,
this
al-
lows
us
to
de-
ter-
mine
whic
op-
er-
a-
tion

failed, making the error message a little less cryptic.

Paramet

- **doc**
The

JSO
doc-
u-
men
to
patc

- **patc**
The
JSO
patc
to
ap-
ply.

Returns

The
re-
sult
of
the
patc
op-
er-
a-
tion.

Raises

Patc
ror
if
the
patc
fails
to
ap-
ply.

Raises

ex-
cep-
tion.
if
the
patc
adds
a
new
root
at-
tribu

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
re-
ques
on
al-
lo-
ca-
tion.

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

al-
lo-
ca-
tion,
the
UUI
or
log-
i-
cal
nam
of
a
node

Raises

HTT
For-
bid-

den
if
the
pol-
icy
for
bids
ac-
cess

Raises

Al-
lo-
ca-
tion-
Not-
Four
if
the
node
is
not
foun

Returns

RPC
node
iden-
ti-
fied
by
node

ironic.a

ironic.a

ironic.a

Che
if
de-
ploy
step
are
al-
lowe

ironic.a

Che
if
get-

ting
de-
taile
driv
info
is
al-
lowe

Ver-
sion
1.30
of
the
API
al-
lows
this.

`ironic.a`

Che
if
fil-
ter-
ing
node
by
con-
duc-
tor
is
al-
lowe

Ver-
sion
1.49
of
the
API
al-
lows
fil-
ter-
ing
node
by
con-
duc-
tor.

`ironic.a`

Che
if
fil-
ter-
ing
node
by
con-
duc-
tor_
is
al-
lowe

Ver-
sion
1.46
of
the
API
al-
lows
fil-
ter-
ing
node
by
con-
duc-
tor_
ironic.a

ironic.a

Che
if
fil-
ter-
ing
node
by
fault
is
al-
lowe

Ver-
sion
1.42
of
the
API
al-

lows
fil-
ter-
ing
node
by
fault

ironic.a

Che
if
fil-
ter-
ing
node
by
lesse
is
al-
lowe

Ver-
sion
1.62
of
the
API
al-
lows
fil-
ter-
ing
node
by
lesse

ironic.a

Che
if
fil-
ter-
ing
node
by
own
is
al-
lowe

Ver-
sion
1.50

of
the
API
al-
lows
fil-
ter-
ing
node
by
own

ironic.a

Che
if
fil-
ter-
ing
drive
by
clas-
sic/c
is
al-
lowe

Ver-
sion
1.30
of
the
API
al-
lows
this.

ironic.a

ironic.a

Che
if
fil-
ter-
ing
node
by
drive
is
al-
lowe
Ver-

sion
1.16
of
the
API
al-
lows
fil-
ter
node
by
driv

ironic.a

Che
if
fetch
ing
a
sub-
set
of
the
re-
sour
at-
tribu
is
al-
lowe

Ver-
sion
1.8
of
the
API
al-
lows
fetch
ing
a
sub-
set
of
the
re-
sour
at-
tribu
this

method checks if the required version is being requested.

ironic.a

Che
if
fil-
ter-
ing
node
by
re-
sour
is
al-
lowe

Ver-
sion
1.21
of
the
API
al-
lows
fil-
ter-
ing
node
by
re-
sour

ironic.a

Che
if
fetc
ing
a
par-
tic-
u-
lar
field
is
al-
lowe

This
meth
chec
if
the

lowed to be fetched in a particular API version.

re-
quir
ver-
sion
is
be-
ing
re-
ques
for
field
that
are
only
al-

ironic.a

Che
if
fetc
ing
a
par-
tic-
u-
lar
field
of
a
port
grou
is
al-
lowe

This
meth
chec
if
the
re-
quir
ver-
sion
is
be-
ing
re-
ques
for

lowed to be fetched in a particular API version.

field
that
are
only
al-

ironic.a

Che
for
re-
ques
non-
exist
field

Che
if
the
user
re-
ques
non-
exist
field

**Paramet
fie.**

A
list
of
field
re-
ques
by
the
user

Object_f

A
list
of
field
sup-
port
by
the
ob-
ject.

Raises

In-
valic

Pa-
ram-
e-
ter-
Valu
if
in-
valid
field
were
re-
ques

ironic.a

Che
if
fil-
ter-
ing
node
by
pro-
vi-
sion
state
is
al-
lowe

Ver-
sion
1.9
of
the
API
al-
lows
fil-
ter
node
by
pro-
vi-
sion
state

ironic.a

Che
if
the
list

pol-
icy
au-
tho-
rizes
this
re-
ques
on
an
ob-
ject.

Param

ob-
ject
type
of
ob-
ject
be-
ing
check

Param

own
own
fil-
ter
for
list
quer
if
any

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou

be
used
for
list
quer
if
need

ironic.a

Che
if
the
spec
i-
fied
poli-
cies
au-
tho-
rize
this
re-
ques
on
a
node

Param

pol-
icy_
List
of
pol-
icy
nam
to
chec

Param

node
the
UI
or
log-
i-
cal
nam
of
a
node

Param

with
when
the
RPC
node
should
in-
clud
the
suf-
fix

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-
fied
by
node

ironic.a

Che
if
the

spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a
node

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

node
the
UUI
or
log-
i-
cal
nam
of
a
node

Param

with
whe
the
RPC
node
shou
in-
clud
the
suf-
fix

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-
fied
by
node

ironic.a

Che
if
the
pol-
icy
au-
tho-
rizes
this
re-
ques
on
an
ob-
ject.

Param

ob-

ject_
type_
of
ob-
ject
be-
ing
chec

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

own
the
own

Param

lesse
the
lesse

Param

con-
ceal
the
UUI
of
the
node
IF
we
shou
con-
ceal
the
ex-
is-
tenc
of
the
node

with a 404 Error instead of a 403 Error.

Raises

HTT

For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
is
au-
tho-
rise
for
this
re-
ques

Policy_n

Nam
of
the
pol-
icy
to
chec

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
is
au-
tho-
rise
for
this
re-
ques

Policy_n

Nam
of
the
pol-
icy
to
chec

Returns

True
if
pol-
icy
is
mat
oth-
er-
wise
false

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-

rizes
this
re-
ques
on
a
port

Paramet

- **por**
Boo
valu
de-
fault
false
in-
di-
cat-
ing
if
the
list
pol-
icy
chec
is
for
a

portgroup as the policy names are different between ports and portgroups.

- **par**
The
UUI
of
a
node
if
any,
to
ap-
ply
a
pol-
icy
chec
to
as
well
be-

fore applying other policy check operations.

/v1/portgroups/<uuid>/ports.

•

par
The
UUID
of
the
par-
ent
port
group
if
the
list
of
port
was
re-
triev
via
the

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou
be
used
for
list
quer
if
need

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a
port

Param
pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param
port.
The
nam
uuid
or
othe
valid
ID
valu
to
find
a
port
or
port
grou
by.

Raises
HTT
For-
bid-

den
if
the
pol-
icy
for
bids
ac-
cess

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Returns

RPC
port
iden
ti-
fied
by
port
as-
so-
ci-
ated
node

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a

fore applying other policy check operations.

vol-
ume
Parameter
parameter
The
UI
of
a
node
if
any,
to
ap-
ply
a
pol-
icy
check
to
as
well
be-

Raises
HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Returns
own
that
shou
be
used
for
list
quer
if
need

ironic.a

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a
vol-
ume

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

vol_
The
nam
uuid
or
othe
valid
ID
valu
to
find
a
vol-
ume
tar-
get
or
con-

tor by.

connector. Default value is False, implying connector.

nec-
Param
tar-
get:
Boo
valu
to
in-
di-
cate
if
the
chec
is
for
a
vol-
ume
tar-
get
or

Raises
HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Raises
Vol-
ume
Con
nec-
torN
Four
if
the
node
is
not
foun

Raises

Vol-
ume
get-
Not-
Four
if
the
node
is
not
foun

Returns

RPC
port
iden
ti-
fied
by
port
as-
so-
ci-
ated
node

`ironic.a`

Gen
er-
a-
tor
of
field
not
al-
lowe
in
the
cur-
rent
re-
ques

`ironic.a`

Get
re-
serv
nam
for
a

that can not be used as an identifier for a resource because the names are either being used as a custom action or is the name of a nested controller inside the given class.

give
con-
troll

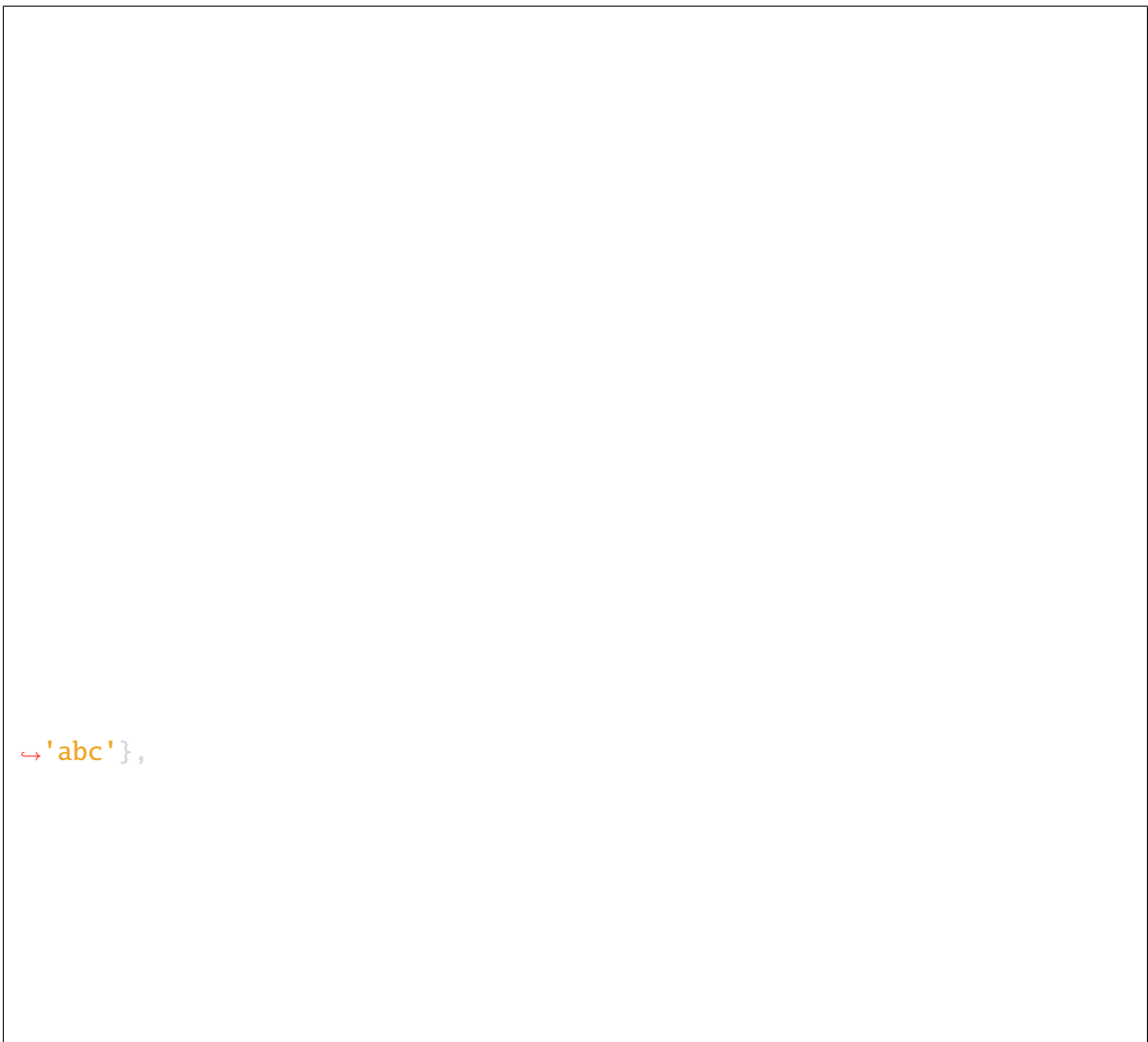
In-
spec
the
con-
troll
class
and
re-
turn
the
re-
serv
nam
with
it.
Re-
serv
nam
are
nam

Paramet
cls
The
con-
troll
class
to
be
in-
spec

ironic.a

Get
the
patc
val-
ues
cor-
re-
spor
ing
to
the
spec
i-

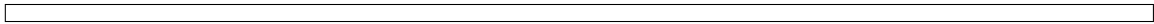
field
path
If
there
are
mul-
ti-
ple
val-
ues
spec-
i-
fied
for
the
sam-
e path
for
ex-
am-
ple



↔ 'abc' },

(continues on next page)

(continued from previous page)



re-
turn
all
of
then
in
a
list
(pre-
serv
ing
or-
der)

Paramet

- **pat**
HTT
PAT
re-
ques
body
- **patl**
the
path
to
get
the
patc
val-
ues
for.

Returns

list
of
val-
ues
for
the
spec
i-
fied
path
in
the

patc
ironic.a

Cal-
cu-
late
field
to
re-
turn
from
an
API
re-
ques

The
field
quer
and
de-
tail=
quer
can
not
be
pass
into
a
re-
ques
at
the
sam
time
To

use the detail query we need to be on a version of the API greater than expected, likewise some APIs require a certain version for the fields query. This function raises an `InvalidParameterValue` exception if any of these conditions are not met.

If
thes
chec
pass
then
this
func
tion
will
re-

fault fields provided.

turn
ei-
ther
the
field
pass
in
or
the
de-

Paramet

- **fi:**
The
field
quer
pass
into
the
API
re-
ques
- **de:**
The
de-
tail
quer
pass
into
the
API
re-
ques
- **def:**
The
de-
fault
field
to
re-
turn
if
field
and
de-

tail=

- **che**
Func
tion
to
chec
if
de-
tail
quer
is
al-
lowe
base
on
the
ver-
sion

- **che**
Func
tion
to
chec
if
field
quer
is
al-
lowe
base
on
the
ver-
sion

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
there
is
an
in-
valic

strings or API version.

com
bi-
na-
tion
of
quer

Returns

field
pass
in
valu
or
de-
fault

ironic.a

Get
the
RPC
al-
lo-
ca-
tion
from
the
al-
lo-
ca-
tion
UUID
or
log-
i-
cal
nam

Paramet

all
the
UUID
or
log-
i-
cal
nam
of
an
al-
lo-
ca-

tion.

Returns

The
RPC
al-
lo-
ca-
tion.

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Al-
lo-
ca-
tion-
Not-
Foun
if
the
al-
lo-
ca-
tion
is
not
foun

ironic.a

Get
the
RPC
al-
lo-
ca-
tion
from

tion_ident with .json suffix. Otherwise identical to get_rpc_allocation.

the
al-
lo-
ca-
tion
UUI
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
al-
lo-
ca-

Parameter

all
the
UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

Returns

The
RPC

al-
lo-
ca-
tion.

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Al-
lo-
ca-
tion-
Not-
Foun
if
the
al-
lo-
ca-
tion
is
not
foun

ironic.a

Get
the
RPC
de-
ploy
tem-
plate
from
the
UI
or
log-
i-

cal
nam

Parameter

temp
the
UI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

Returns

The
RPC
de-
ploy
tem-
plate

Raises

In-
valid
uidC
Name
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-

plate
is
not
foun

ironic.a

Get
the
RPC
de-
ploy
tem-
plate
from
the
UI
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
tem-
plate
with

.json suffix. Otherwise identical to get_rpc_deploy_template.

Paramet
temp
the
UI
or
log-
i-
cal

nam
of
a
de-
ploy
tem-
plate

Returns

The
RPC
de-
ploy
tem-
plate

Raises

In-
valid
uidC
Name
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
is
not
found

ironic.a

Get
the
RPC
node

from
the
node
uuid
or
log-
i-
cal
nam

Paramet

nod
the
UUID
or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Nod

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Nod
Not-
Four
if
the
node
is
not

found
ironic.a

Get
the
RPC
node
from
the
node
uuid
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
node
with
.json

suffix. Otherwise identical to `get_rpc_node`.

Paramet
node
the
UUID
or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Nod

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

ironic.a

Get
the
RPC
port
grou
from
the
port
grou
UU
or
log-
i-
cal
nam

Paramet

por
the

UU
or
log-
i-
cal
nam
of
a
port
grou

Returns

The
RPC
port
grou

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Port
grou
Not-
Foun
if
the
port
grou
is
not
foun

ironic.a

Get
the
RPC
port
grou

.json suffix. Otherwise identical to `get_rpc_portgroup`.

from
the
port
group
UUID
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
port
group
with

Parameter

port
the
UUID
or
log-
i-
cal
nam
of
a
port
group

Returns

The
RPC
port
group

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Port
grou
Not-
Foun
if
the
port
grou
is
not
foun

ironic.a

Re-
turn
node
state
to
use
by
de-
fault
whe
cre-
at-
ing
new
node

Pre-
vi-
ousl
the
de-
fault

ENROLL.

state
for
new
node
was
AVA
ABI
Star
ing
with
API
1.11
it
is

ironic.a

Re-
turn
when
the
patch
in-
clud
re-
mov
of
the
path
(or
sub-
path
of).

Parameter

- **patch**
HTT
PAT
re-
ques
body
- **path**
the
path
to
check

Returns

True
if
path
or
sub-
path
be-
ing
re-
mov
Fals
oth-
er-
wise

ironic.a

Re-
turn
whe
the
patc
in-
clud
op-
er-
a-
tion
on
path
(or
its
sub-
path

Paramet

- **pat**
HTT
PAT
re-
ques
body
- **patl**
the
path
to
chec

Returns

True
if
path
or
sub-
path
be-
ing
patc
Fals
oth-
er-
wise

`ironic.a`

De-
ter-
mine
if
the
pro-
vide
nam
is
a
valid
host
nam

`ironic.a`

De-
ter-
mine
if
the
pro-
vide
nam
is
a
valid
node
nam

Che
to
see
that
the
pro-
vide

node
nam
is
valid
and
isnt
a
UUID

Paramet

nam
the
node
nam
to
chec

Returns

True
if
the
nam
is
valid
False
oth-
er-
wise

ironic.a

ironic.a

Help
func
tion
to
con-
vert
RPC
ob-
jects
to
RES
API
dicts

Paramet

-

obj
RPC
ob-
ject
to
con-
vert
to
a
dict

- **inc:**
When
to
in-
clud
stan-
dard
base
class
at-
tribu
cre-
ated

- **inc:**
When
to
in-
clud
stan-
dard
base
class
at-
tribu
up-
date

- **inc:**
When
to
in-
clud
stan-
dard
base
class
at-
tribu

resource name

the object `uuid` will be used.

uuid

- **link**
When
spec
i-
fied,
gen-
er-
ate
a
link
valu
with
a
sel.
and
book
us-
ing
this

- **link**
Re-
sour
ar-
gu-
men
to
be
add
to
gen-
er-
ated
link
When
not
spec
i-
fied,

- **file**
Key
nam
for
dict
val-

ues
to
pop-
u-
late
di-
rectl
from
ob-
ject
at-
tribu

Returns

A
dict
con-
tain-
ing
val-
ues
from
the
ob-
ject

ironic.a

Up-
date
rpc
ob-
ject
base
on
char
field
in
a
dict.

Only
field
whic
have
a
cor-
re-
spon
ing
sche
field

be updated using the `id_map`.

are
up-
date
whe
char
Othe
val-
ues
can

Parameter

- **from**
Dict
con-
tain-
ing
char
field
val-
ues
- **rpc**
Ob-
ject
to
up-
date
char
field
on
- **field**
Field
nam
on
the
rpc
ob-
ject
- **schema**
json
sche
to
get
field
nam

no matching field in the schema

of
the
dict
•
id_
Op-
tion:
dict
map
ping
ob-
ject
field
nam
to
ar-
bi-
trary
val-
ues
whe
there
is

ironic.a

Val-
i-
date
that
a
patc
list
only
mod
i-
fies
al-
lowe
field

Paramet

•
pat
List
of
patc
dicts
to

val-
i-
date

- **all**
List
of
field
whic
are
al-
lowe
to
be
patc

Returns

The
list
of
field
whic
will
be
patc

Raises

ex-
cep-
tion.
if
any
patc
char
a
field
not
in
all

ironic.a

Val-
i-
date
a
patc
dict
ob-
ject
agai
a

val-
ida-
tor
or
sche

This
func
tion
has
the
side
effe
of
dele
ing
any
dict
valu
whic
is
not
in
the
sche
This

allows database-loaded objects to be pruned of their internal values before validation.

Paramet

- **pat**
dict
rep-
re-
sen-
ta-
tion
of
the
ob-
ject
with
patc
up-
date
ap-
plie
- **sch**
Any

is specified then the resulting `patched_dict` will be validated against the schema

beyond the schema

dict
key
not
in
the
sche
will
be
dele
from
the
dict.
If
no
val-
ida-
tor

- **val.**
Op-
tion:
val-
ida-
tor
to
use
if
there
is
ex-
tra
val-
i-
da-
tion
re-
quir

Raises
ex-
cep-
tion.
if
val-
i-
da-
tion
fails

ironic.a

Look
up
the
node
ref-
er-
ence
in
the
ob-
ject
and
pop-
u-
late
a
dict.

The
node
is
fetch
with
the
ob-
ject
node
at-
tribu
and
the
dict
node
valu
is
pop-
u-
latec

with the node uuid

Parameter

- **obj**
ob-
ject
to
get
the
node
at-

tribu

- **to_**
dict
to
pop-
u-
late
with
a
nod
valu

Raises

ex-
cep-
tion.
if
the
nod
is
not
foun

ironic.a

Re-
plac
nod
dict
valu
with
nod

nod
is
foun
by
fetc
ing
the
nod
by
id
look

Paramet

to_
Dict
to
set
nod
valu

on

Returns

The node object from the look

Raises

NodeNotFound
FourDigitStatusSetError
to 400 Bad Request
when node is not found

ironic.a

Replaces node dict value with node

node is found by fetching the node by uuid look

Parameter

to_
Dict
to

set
node
valu
on

Returns

The
node
ob-
ject
from
the
look

Raises

Nod
Not-
Four
with
sta-
tus_
set
to
400
BAD
whe
node
is
not
foun

ironic.a

Re-
mov
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in

the
fie
pa-
ram-
e-
ter
(plu
the
lin
field

Paramet

- **to_**
dict
to
san-
i-
tize
- **fie**
(*lis*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
then
all

ironic.a

ironic.a

ironic.a

Call
a
ven-
dor
pass
API

return code for methods that are asynchronous or synchronous; Attach the return value to the response object if its being served statically.

for drivers vendor passthru this is the drivers name.

ex-
ten-
sion

Call
the
ven-
dor
pass
API
ex-
ten-
sion
and
pro-
cess
the
meth
re-
spor
to
set
the
righ

Paramet

- **iden**
The
re-
sour
iden
ti-
fi-
ca-
tion.
For
node
ven-
dor
pass
this
is
the
node
UUI

•

metl

The
ven-
dor
meth
nam

•

top

The
RPC
topic

•

dat

The
data
pass
to
the
ven-
dor
meth
De-
fault
to
Non

•

dri

Boo
valu
Whe
this
is
a
node
or
driv
ven-
dor
pass
De-
fault
to
Fals

Returns

A
WSI
re-
spor
ob-

ject
to
be
re-
turn
by
the
API

ironic.api.controllers.v1.versions module

ironic.a

Re-
turn
the
max
i-
mun
sup-
port
API
ver-
sion
(as
a
strin

If
the
ser-
vice
is
pinn
the
max
i-
mun
API
ver-
sion
is
the
pinn
ver-
sion
Oth-
er-

wise, it is the maximum supported API version.

ironic.a

Re-
turn
the
min-
i-
mun
sup-
port
API
ver-
sion
(as
a
strin

`ironic.api.controllers.v1.volume module`

class ir

Base
pec
res
Res

RES
con-
troll
for
vol-
ume
root

get()

`ironic.a`

`ironic.api.controllers.v1.volume_connector module`

class ir

Base
pec
res
Res

RES
con-
troll
for

Vol-
ume
Con
nec-
tors.

delete()

Dele
a
vol-
ume
con-
nec-
tor.

Parame

con
UUI
of
a
vol-
ume
con-
nec-
tor.

Raises

Op-
er-
a-
tion)
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Nod
Loc
if

node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

Raises

In-
valid

powered off.

State
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
con-
nec-
tor
is
not

get_all

Re-
triev
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **node**
UUID
or
name
of
a
node
to
get
only
vol-
ume
con-
nec-
tors
for
that

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

node

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This
- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.
- **sor**
di-

returned.

rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file:**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **det:**
Op-
tion:
whe
to
re-
triev
with
de-
tail.

Returns
a
list
of
vol-
ume
con-
nec-

is found.

tors,
or
an
emp
list
if
no
vol-
ume
con-
nec-
tor

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
key
is
in-
valid
for
sort-
ing.

Raises

In-
valid
Pa-
ram-
e-

ter-
Valu
if
both
field
and
de-
tail
are
spec
i-
fied.

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
vol-
ume
con-
nec-
tor.

Parame

- **con**
UUI
of
a
vol-
ume
con-
nec-
tor.
- **fie**
Op-
tion
a
list
with
a
spec

returned.

i-
fied
set
of
field
of
the
re-
sour
to
be

Returns

API
seria
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Vol-
ume
Con
nec-
torN
Four
if

fed UUID.

no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-

invalid.

patch(cc

Up-
date
an
ex-
ist-
ing
vol-
ume
con-
nec-
tor.

Paramete

- **con**
UUID
of
a
vol-
ume
con-
nec-
tor.
- **pat**
a
json
PAT
doc-
u-
men
to

ap-
ply
to
this
vol-
ume
con-
nec-
tor.

Returns

API
serial
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Patc
ror
if
a
give
patc
can
not
be
ap-

fied UUID.

plied
Raises
Vol-
ume
Con
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
con-
nec-
tors
UUI
is
be-
ing
char

Raises
Nod
Lock
if
node
is
lock
by

an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
an-
othe
con-
nec-
tor
al-
read

exists with the same values for type and connector_id fields

Raises

In-
valid
UID

powered off.

if
in-
valid
node
UUID
is
pass
in
the
patc

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
con-
nec-
tor
is
not

post(*con*)

Cre-
ate
a
new
vol-
ume
con-
nec-
tor.

Parame

con
a
vol-
ume
con-
nec-
tor

with
the
re-
ques
body

Returns

API
seria
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
a
vol-

ready exists with the same type and connector_id

ready exists

ironic.api.controllers.v1.volume_target module

ume
con-
nec-
tor
al-

Raises

Vol-
ume
Con
nec-
torA
read
ists
if
a
vol-
ume
con-
nec-
tor
with
the
sam
UUI
al-

ironic.a

ironic.a

class ir

Base
pec
res
Res
RES
con-
troll

for
Vol-
ume
gets.

delete()

Dele
a
vol-
ume
tar-
get.

Parame

tar
UUI
of
a
vol-
ume
tar-
get.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Nod
Lock
if
node
is
lock

by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
tar-
get
does
not
ex-
ist

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

In-
valid
State
If
a
node
as-
so-

ered off.

ci-
ated
with
the
vol-
ume
tar-
get
is
not
pow

get_all

Re-
triev
a
list
of
vol-
ume
tar-
gets.

Parame

- **node**
UI
or
nam
of
a
node
to
get
only
vol-
ume
tar-
gets
for
that
node
- **mar**
pag-
i-
na-
tion

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc
De-

returned.

fault
asc.

- **file:**
Optionally, you can specify a list with a specified set of fields of the resource to be

- **detail:**
Optionally, you can specify whether to retrieve details.

- **project:**
Optionally, you can specify an associated node project (owner) or less than one file.

ter
the
quer
upon

Returns

a
list
of
vol-
ume
tar-
gets.
or
an
emp
list
if
no
vol-
ume
tar-
get
is
foun

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
key
is

in-
valid
for
sort-
ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
both
field
and
de-
tail
are
spec
i-
fied.

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
vol-
ume
tar-
get.

Parame

- **tar**
UI
of
a
vol-
ume
tar-
get.

returned.

-

file
Optionally, a list with a specified i-fied set of fields of the resource to be

Returns

API serial volume target object.

Raises

OperationPermitted if access with specifying a parent node

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
UUI
ex-
ists

invalid

patch(*ta*

Up-
date
an
ex-
ist-
ing
vol-
ume
tar-
get.

Parame

- **tar**
UUI
of
a
vol-
ume
tar-
get.
- **pat**
a
json
PAT
doc-
u-

men
to
ap-
ply
to
this
vol-
ume
tar-
get.

Returns

API
serial
vol-
ume
tar-
get
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Patc
ror
if
a
give
patc
can
not
be
ap-

plied

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
tar-
gets
UU
is
be-
ing
char

Raises

Nod
Lock
if
the
node
is
al-
read
lock

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
does
not
ex-

ist

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

the same node ID and boot index values

Raises

In-
valid
UID
if
in-
valid
node
UID
is
pass
in

ered off.

the
patc

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
is
not
pow

post(*tar*

Cre-
ate
a
new
vol-
ume
tar-
get.

Paramete

tar
a
vol-
ume
tar-
get
with
the
re-
ques
body

Returns

API
seria
vol-

ume
tar-
get
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

the same node ID and boot index

Raises

Vol-
ume
ge-

tAl-
read
ists
if
a
vol-
ume
tar-
get
with
the
sam
UU
ex-
ists

ironic.a

ironic.a

Module contents

Ver-
sion
1
of
the
Iron
API
Spe
i-
fi-
ca-
tion
can
be
foun
at
doc/

class ir
Base
obj
Ver-
sion

1
API
con-
troll
root

index()

Submodules

ironic.api.controllers.base module

class ir

Base
obj
API
Ver-
sion
ob-
ject.

max_str

HTT
re-
spor
head

min_str

HTT
re-
spor
head

static p

De-
ter-
min
the
API
ver-
sion
re-
ques
base
on
the
head

ers
sup-
plie

Paramete

- **hea**
we-
bob
head
ers

- **def**
ver-
sion
to
use
if
not
spec
i-
fied
in
head
ers

- **lat**
ver-
sion
to
use
if
lat-
est
is
re-
ques

Returns

a
tu-
ple
of
(ma-
jor,
mi-
nor)
ver-
sion
num

bers

Raises

we-

bob.

string =

HTT

Hea

strin

car-

ry-

ing

the

re-

ques

ver-

sion

ironic.api.controllers.link module

ironic.a

ironic.a

Buil

a

dict

rep-

re-

sent

ing

a

link

ironic.api.controllers.root module

class ir

Base

obj

index(*c

ironic.a

ironic.api.controllers.version module

ironic.a

ironic.a

Re-
turn
a
dict
rep-
re-
sent
ing
the
cur-
rent
de-
fault
ver-
sion

id:
The
ID
of
the
(ma-
jor)
ver-
sion
also
acts
as
the
re-
lease
num-
ber
link.
A
list

containing one link that points to the current version of the API

sta-
tus:
Sta-
tus
of
the
ver-
sion

one
of
CUE
REN
SUP
POR
DEF
RE-
CAT
min.
The
cur-
rent.
max
i-
mun
sup-
port
(ma-
jor.n
ver-
sion
of
API
ver-
sion
Min
i-
mun
sup-
port
(ma-
jor.n
ver-
sion
of
API

Module contents

`ironic.api.middleware` package

Submodules

`ironic.api.middleware.auth_public_routes` module

`class ir`

Base

obj

A

wrap

per

on

au-

then

ti-

ca-

tion

mid-

dle-

ware

Doe

not

per-

form

ver-

i-

fi-

ca-

tion

of

au-

then

ti-

ca-

tion

to-

kens

for

pub-

lic

routes in the API.

`ironic.api.middleware.json_ext` module

class ir

Base
obj

Sim-
pli-
fied
pro-
cess
ing
of
.json
ex-
ten-
sion

Pre-
vi-
ously
Iron
API
used
the
gues-
sured
fea-
ture.
It
was
never
needed
as
we
never
al-
lowed
non-

JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

`ironic.api.middleware.parsable_error` module

Mid-
dle-
ware
to
re-
plac
the

formatted so the client can parse it.

Module contents

plain
text
mes
sage
body
of
an
er-
ror
re-
spor
with
one

Base
on
peca

class ir

Base
obj
Re-
plac
er-
ror
body
with
som
thing
the
clien
can
pars

class ir

Base
obj
A
wrap
per
on
au-
then
ti-
ca-

routes in the API.

tion
mid-
dle-
ware

Doe
not
per-
form
ver-
i-
fi-
ca-
tion
of
au-
then-
ti-
ca-
tion
to-
kens
for
pub-
lic

class ir

Base
obj

Sim
pli-
fied
pro-
cess
ing
of
.json
ex-
ten-
sion

Pre-
vi-
ousl
Iron
API
used
the
gues
fea-

JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

Submodules

ironic.api.app module

ture.
It
was
neve
need
as
we
neve
al-
lowe
non-

class ir

Base
obj
Re-
plac
er-
ror
body
with
som
thing
the
clie
can
pars

class ir

Base
osl
cor
COR
Iron
spec
COE
class
Wer
addi
the
Iron
spec

that a request bearing those headers might be accepted by the Ironic REST API.

ironic.api.config module

ironic.api.functions module

ver-
sion
head
ers
to
the
list
of
sim-
ple
head
ers
in
or-
der

```
simple_  
'Cache-  
'X-Auth-  
'X-OpenS  
'X-OpenS
```

```
class ir  
    Base  
    obj  
ironic.a  
  
ironic.a
```

```
class ir  
  
    Base  
    obj  
  
    An  
    ar-  
    gu-  
    men  
    def-  
    i-  
    ni-  
    tion  
    of  
    an  
    api
```

en-
try

datatype
Data
type

default
De-
fault
valu
if
ar-
gu-
men
is
omit
ted

mandator
True
if
the
ar-
gu-
men
is
man
tory

name
ar-
gu-
men
nam

resolve

class ir
Base
obj
An
api
en-
try
def-
i-
ni-
tion

argument

The
func
tion
ar-
gu-
men
(list
of
Fun

body_type

If
the
body
carr
the
data
of
a
sin-
gle
ar-
gu-
men
its
type

doc

Func
tion
doc-
u-
men
ta-
tion

extra_options

Dic-
tio-
n-
nary
of
prot
spec
op-
tions

static_generator

Re-
turn
the
Fun

function and not raise `UnknownArgument` exceptions

of
a
meth

get_arg

Re-
turn
a
Fun
from
its
nam

ignore_e

True
if
ex-
tra
ar-
gu-
men
shou
be
ig-
nore
NOT
in-
serte
in
the
kwa
of
the

name

Func
tion
nam

resolve_

return_t

Re-
turn
type

set_arg

set_opt:

status_c

Sta-
tus
code

ironic.a

ironic.a

ironic.a
alias
of
ironic
api
fun
sig

class ir

Base
obj

Dec
o-
ra-
tor
that
spec
ify
the
ar-
gu-
men
type
of
an
ex-
pose
func
tion.

Paramet

- **ret**
Type
of
the
valu
re-
turn
by
the

request body by itself, its type.

func
tion

- **arg**
Type
of
the
Nth
ar-
gu-
men

- **body**
If
the
func
tion
take
a
fi-
nal
ar-
gu-
men
that
is
sup-
pose
to
be
the

- **sta**
HTT
re-
turn
sta-
tus
code
of
the
func
tion.

- **ign**
Al-
low
ex-

you are not using WSME on top of another framework.

rator, either a new decorator named `@wsexpose` that takes the same parameters (it will in addition

tra/u
ar-
gu-
men
(de-
fault
to
Fals
Mos
of
the
time
this
dec-
o-
ra-
tor
is
not
sup-
pose
to
be
used
di-
rectl
un-
less
If
an
adap
is
used
it
will
pro-
vide
ei-
ther
a
spe-
ciali
ver-
sion
of
this
decc
ro-

expose the function, hence its name).

ironic.api.hooks module

ironic.a

class ir

Base

pec

hool

Pec

At-

tach

the

con-

fig

ob-

ject

to

the

re-

ques

so

con-

troll

can

get

to

it.

before (s

Over

ride

this

meth

to

cre-

ate

a

hool

that

gets

call

af-

ter

rou

ing,

but

request gets passed to your controller.

be-
fore
the

Parameter

state
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class inheritance

Base
pec
hool
Peca

Con
fig-
ures
a
re-
ques
con-
text
and
at-
tach
it
to
the
re-
ques

after(*state*)

Ove
ride
this
meth
to
cre-
ate
a

dled by the controller.

request gets passed to your controller.

hool
that
gets
called
af-
ter
the
re-
ques
has
been
han-

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

before (s

Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
called
af-
ter
rout
ing,
but
be-
fore
the

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class ir
Base
pec
hool
Pec

At-
tach
the
dbap
ob-
ject
to
the
re-
ques
so
con-
troll
can
get
to
it.

before (s

Over
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

request gets passed to your controller.

ter
rout
ing,
but
be-
fore
the

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class ir

Base
pec
hool
Pec

Wor
rpc.
de-
se-
ri-
al-
ize_

de-
se-
ri-
al-
ize_
buil
rpc
ex-
cep-
tion
trac
back
into
er-
ror
mes

sent to the client. Such behavior is a security concern so this hook is aimed to cut-off traceback from the error message.

dled by the controller.

sage
whic
is
then

after(*sta*

Over
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
the
re-
ques
has
been
han-

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class ir

Base
pec
hool
Pec
At-
tach

API service is behind a proxy or SSL terminator.

the
righ
pub-
lic_u
to
the
re-
ques

At-
tach
the
righ
pub-
lic_u
to
the
re-
ques
so
re-
sour
can
cre-
ate
links
even
whe
the

before (

Over
ride
this
meth
to
cre-
ate
a
hool
that
gets
call
af-
ter
rout
ing,
but
be-
fore

request gets passed to your controller.

the

Parameter

state

The

Peca

sta

ob-

ject

for

the

cur-

rent

re-

ques

class ir

Base

pec

hool

Peca

At-

tach

the

rp-

cap

ob-

ject

to

the

re-

ques

so

con-

troll

can

get

to

it.

before (s

Over

ride

this

meth

to

cre-

ate

a

request gets passed to your controller.

ironic.api.method module

hool
that
gets
called
af-
ter
rout
ing,
but
be-
fore
the

Parameter

state
The
Peca
state
ob-
ject
for
the
cur-
rent
re-
ques

`ironic.a`

`ironic.a`

Dec
o-
ra-
tor
whic
plac
HTT
re-
ques
body
JSO
into
a
meth
ar-
gu-
men

Parameter body:
Name of argument to populate with body JSON

`ironic.a`

`ironic.a`

Extract information that can be sent to the client

`ironic.api.wsgi` module

WSO script for Iron API installation by pbr.

`ironic.a`

Module contents

ironic.cmd package

Submodules

ironic.cmd.api module

The
Iron
Ser-
vice
API

ironic.c

ironic.cmd.conductor module

The
Iron
Man-
age-
men
Ser-
vice

ironic.c

ironic.c

ironic.c

ironic.cmd.dbsync module

Run
stor-
age
data
mi-
gra-
tion.

class ir
Base

obj

check_ob

Che
the
ver-
sion
of
ob-
ject

Che
that
the
ob-
ject
ver-
sion
are
com
pat-
i-
ble
with
this
re-
lease
of
iron
It
does

this by comparing the objects .version field in the database, with the expected versions of these objects.

Re-
turn
Non
if
com
pat-
i-
ble;
a
strin
de-
scrib
ing
the
is-
sue
oth-
er-

wise
create_s
online_c
revision
stamp()
upgrade
version
ironic.c
ironic.c

ironic.cmd.status module

class ir
Base
osl
upg
Upg
Up-
grad
chec
for
the
iron
statu
up-
grad
chec
com
man
Up-
grad
chec
shou
be
adde
as
sep-
a-
rate

tuple.

Module contents

ironic.common package

Subpackages

ironic.common.glance_service package

Submodules

ironic.common.glance_service.image_service module

meth
ods
in
this
class
and
adde
to
_up-
grad

ironic.c

class ir

Base
obj

call(*me*

Call
a
glan
clien
meth

If
we
get
a
con-
nec-
tion
er-
ror,

retry
the
re-
ques
ac-
cord
ing
to
CON

Paramete

- **metl**
The
metl
re-
ques
to
be
calle
- **arg**
A
list
of
po-
si-
tiona
ar-
gu-
men
for
the
metl
calle
- **kwa**
A
dict
of
key-
wor
ar-
gu-
men
for
the
metl
calle

Raises

Glance
Con-
nec-
tion-
Fail

download

Call
out
to
Glance
for
data
and
write
data

Parameters

- **image**
The
opaque
image
identifier.

- **data**
(Optional)
File
object
to
write
data
to.

show(image)

Re-
turn
a
dict
with
image
data

for
the
give
opac
im-
age
id.

Parame

ima
The
opac
im-
age
iden
ti-
fier.

Returns

A
dict
con-
tain-
ing
im-
age
meta
data

Raises

Im-
ageN
Four

Raises

Imag
U-
nac-
cept
able
if
the
im-
age
sta-
tus
is
not
ac-
tive

swift_t

Gen

the temporary Swift URL using the image id from Glance and the config options: `swift_endpoint_url`, `swift_api_version`, `swift_account` and `swift_container`. The temporary URL will be valid for `swift_temp_url_duration` seconds. This allows Ironic to download a Glance image without passing around an `auth_token`.

er-
ate
a
no-
auth
Swi
tem-
po-
rary
URL

This
func
tion
will
gen-
er-
ate
(or
re-
turn
the
cach
one
if
temp
URL
cach
is
en-
able

Parameter
image_id
The
re-
turn
from
a
GET
re-
ques
to
Glar
for
a
cer-
tain

a dictionary, with keys like name and checksum. See <https://docs.openstack.org/glance/latest/user/glanceapi.html> for examples.

tion.

im-
age_
Sho
be

Returns

A
sign
Swi
URI
from
whic
an
im-
age
can
be
dow
load
with
out
au-
then
ti-
ca-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
Swi
con-
fig
op-
tions
are
not
set
cor-
rectl

Raises

Miss
ing-
Pa-

ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
not
set.

Raises

Imag
U-
nac-
cept
able
if
the
im-
age
info
from
Glan
does
not
have
an
im-
age
ID.

class ir

Bas
tup

url

Alia
for
field
num
ber
0

url_exp:

Alia

for
field
num
ber
1

ironic.c

Cre-
ates
a
glan
clien
if
does
ex-
ists
and
calls
the
func
tion.

`ironic.common.glance_service.service_utils` module

ironic.c

ironic.c

Che
the
im-
age
sta-
tus.

This
check
is
need
in
case
the
Glan
im-
age
is
stuck
in
queu
sta-
tus

on.

or
pend
ing_

ironic.c

Che
im-
age
avai
abil-
ity.

This
chec
is
need
in
case
Nov
and
Glar
are
de-
ploy
with
out
au-
then
ti-
ca-
tion
turn

ironic.c

Pars
an
im-
age
id
from
im-
age
href

Paramet

ima
href
of
an
im-
age

Returns

im-
age
id
pars
from
im-
age_

Raises

Inv
whe
in-
put
im-
age
href
is
in-
valid

ironic.c

Module contents

Submodules

ironic.common.args module

ironic.c

Val-
i-
date
that
ev-
ery
sup-
plie
val-
ida-
tor
pass

The
valu
re-
turn
from
each

val-
ida-
tor
is
pass
as
the
valu
to
the
next
one.

Paramet

- **name**
Name
of
the
ar-
gu-
men

- **valu**
A
valu

Returns

The
valu
trans
form
thro
ev-
ery
sup-
plie
val-
ida-
tor

Raises

The
er-
ror
from
the
first
faile
val-
ida-

for
ironic.c
Val-
i-
date
that
the
valu
is
a
strin
rep-
re-
sent
ing
a
bool

Paramet

- **name**
Name
of
the
ar-
gu-
men
- **valu**
A
strin
valu

Returns

The
bool
rep-
re-
sen-
ta-
tion
of
the
valu
or
Non
if
valu
is

Non
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
can-
not
be
con-
verte
to
a
bool

ironic.c

Re-
turn
a
val-
ida-
tor
func
tion
whic
val-
i-
date
dict
field

Val-
ida-
tors
will
re-
plac
the
valu
with
the
val-
i-
da-
tion

has no validator is ignored. When a key is missing in the value then the corresponding validator will not be run.

a validator function to run on that value

re-
sult.
Any
dict
item
whic

Param

val-
ida-
tors
dict
whe
the
key
is
a
dict
key
to
val-
i-
date
and
the
valu
is

Returns

val-
ida-
tor
func
tion
whic
take
nam
and
valu
ar-
gu-
men

ironic.c

Val-
i-
date
that
the

valu
rep-
re-
sent
an
in-
te-
ger

Paramet

- **name**
Name
of
the
ar-
gu-
men

- **valu**
A
valu
rep-
re-
sent
ing
an
in-
te-
ger

Returns

The
valu
as
an
int,
or
Non
if
valu
is
Non

Raises

In-
valic
Pa-
ram-
e-
ter-

Valu
if
the
valu
does
not
rep-
re-
sent
an
in-
te-
ger

ironic.c

Val-
i-
date
that
the
valu
rep-
re-
sent
a
MA
ad-
dres

Paramet

- **name**
Name
of
the
ar-
gu-
men
- **val**
A
strin
valu
rep-
re-
sent
ing
a
MA

ad-
dres

Returns

The
valu
as
a
nor-
mal-
ized
MA
ad-
dres
or
Non
if
valu
is
Non

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
valic
MA
ad-
dres

ironic.c

Val-
i-
date
that
the
valu
is
a
log-
i-
cal

nam

Paramet

- **nam**
Nam
of
the
ar-
gu-
men

- **valu**
A
log-
i-
cal
nam
strin
valu

Returns

The
valu
or
Non
if
valu
is
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
valid
log-
i-
cal
nam

ironic.c

Val-
i-
date
if
at
least
one
sup-
plie
val-
ida-
tor
pass

Paramet

- **name**
Name
of
the
ar-
gu-
men

- **valu**
A
valu

Returns

The
valu
re-
turn
from
the
first
suc-
cess
ful
val-
ida-
tor

Raises

The
er-
ror
from
the

last
val-
ida-
tor
whe
ev-
ery
val-
i-
da-
tion
fails

ironic.c

Val-
i-
date
a
patc
API
op-
er-
a-
tion

ironic.c

Re-
turn
a
val-
ida-
tor
func
tion
whic
val-
i-
date
the
valu
with
json
sche

Param

sche
dict
rep-
re-

sent
ing
json
sche
to
val
i-
date
with

Returns

val-
ida-
tor
func
tion
whic
take
nam
and
valu
ar-
gu-
men

ironic.c

Val-
i-
date
that
the
valu
is
a
strin

Paramet

- **name**
Nam
of
the
ar-
gu-
men
- **valu**
A
strin
valu

Returns

The
strin
valu
or
Non
if
valu
is
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
strin

`ironic.c`

Val-
i-
date
and
con-
vert
com
de-
lim-
ited
strin
to
a
list.

Paramet

- **name**
Nam
of
the
ar-

gu-
men

- **valu**
A
com
sep-
a-
rate
strin
of
val-
ues

Returns

A
list
of
uniq
val-
ues
(low
case
main
tain-
ing
the
sam
or-
der,
or
Non
if
valu

is None

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
strin

ironic.c

Re-
turn
a
val-
ida-
tor
func-
tion
which
checks
the
valu-
is
one
of
the
type

Param

type
one
or
more
type
to
use
for
the
isin-
stan-
test

Returns

val-
ida-
tor
func-
tion
which
takes
name
and
valu-
ar-
gu-
men

ironic.c

Val-
i-

date
that
the
valu
is
a
UUI

Paramet

- **name**
Name
of
the
ar-
gu-
men

- **valu**
A
UUI
strin
valu

Returns

The
valu
or
Non
if
valu
is
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
valid
UUI

ironic.c

Val-
i-
date
that
the
valu
is
a
UUI
or
log-
i-
cal
nam

Paramet

- **name**
Name
of
the
ar-
gu-
men
- **valu**
A
UUI
or
log-
i-
cal
nam
strin
valu

Returns

The
valu
or
Non
if
valu
is
Non

Raises

In-
valid

name

ironic.common.boot_devices module

Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a
valid
UU
or
log-
i-
cal

ironic.c

Dec
o-
ra-
tor
whic
val-
i-
date
and
trans
form
func
tion
ar-
gu-
men

Map
ping
of
boot
de-
vice
used
whe
re-
ques
ing

nate device.

find the documentation at: <http://linux.die.net/man/1/ipmitool>

the
sys-
tem
to
boot
from
an
al-
ter-

The
op-
tions
pre-
sent
were
base
on
the
IP-
MI-
tool
chas
sis
boot
dev
com
man
You
can

NOT
This
mod
ule
does
not
in-
clud
all
the
op-
tions
from
ip-
mi-
tool
be-
caus
they

make sense in the limited context of Ironic right now.

dont
ironic.c
Boo
into
BIO
setu
ironic.c
Boo
from
CD/
ironic.c
Boo
from
de-
fault
Hard
driv
ironic.c
Boo
from
a
flopp
driv
ironic.c
Boo
from
iSCS
vol-
ume
ironic.c
Boo
from
PXE
boot
ironic.c
Boo
from
de-
fault
Hard
driv
re-
ques
Safe
Mod
ironic.c

ironic.common.boot_modes module

tive firmware interfaces.

Boo
from
Wid
Area
Net-
worl

Map
ping
of
boot
mod
used
whe
re-
ques
ing
the
sys-
tem
to
boot
us-
ing
al-
ter-
na-

The
op-
tions
pre-
sent
were
base
on
the
Red
fish
pro-
to-
col
ca-
pa-
bil-
i-
ties,
spec

ically on the `BootSourceOverrideMode` property.

`ironic.c`
Boo
over
lega
PC
BIO
firm
in-
ter-
face

`ironic.c`
Boo
over
Uni-
fied
Ex-
ten-
si-
ble
Firm
In-
ter-
face
(UE
firm
in-
ter-
face

`ironic.common.cinder` module

`ironic.c`
At-
tach
vol-
ume
to
a
node
Enu-
mer-
ate
thro
the
pro-
vide
list

defined in the task utilizing the provided connector information.

attach the volume. If use of the volume fails, a user may need to remove any lingering pre-existing/unused attachment records since we have no way to validate if the connector profile data differs from what was provided to cinder.

Parameter

- **task**
Task
ager
in-
stan-
rep-
re-
sent
ing
the
op-

er-
a-
tion.

- **vol**
List
of
vol-
ume
UUI
val-
ues
rep-
re-
sent
ing
vol-
ume

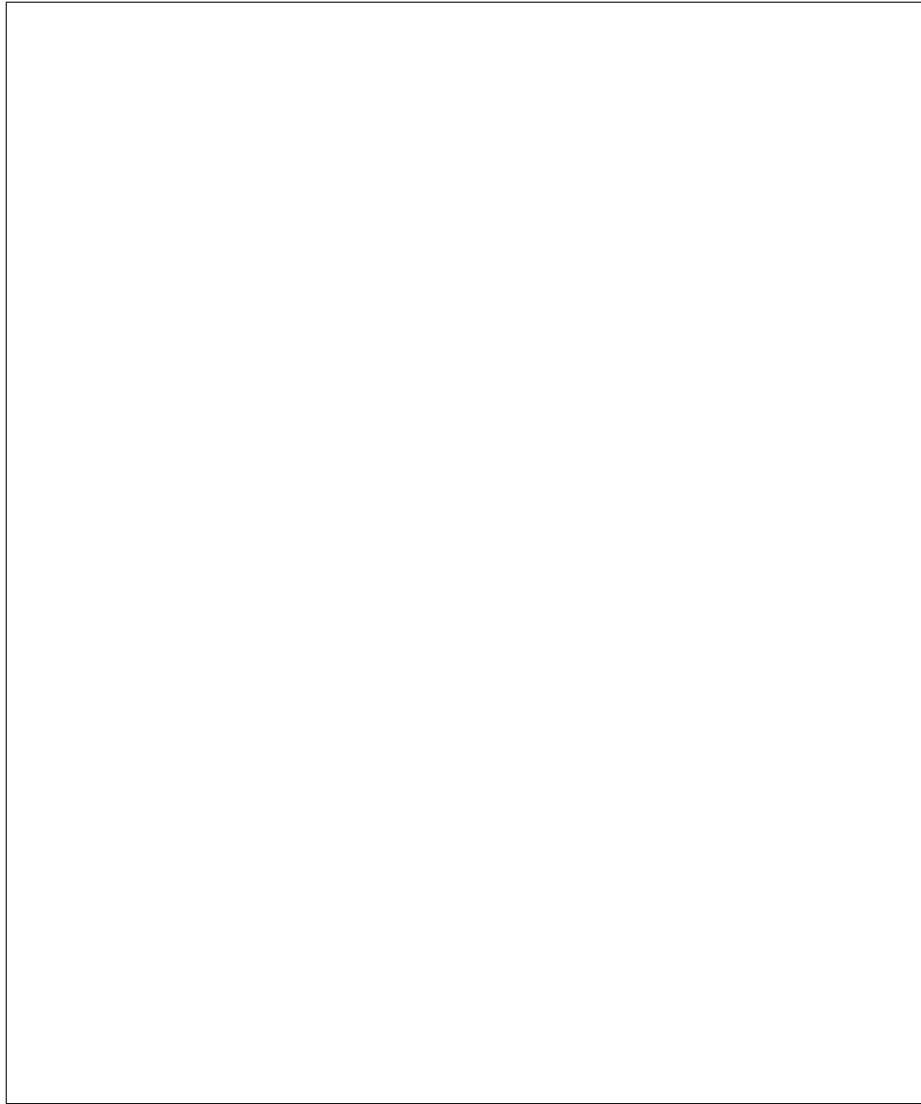
- **con**
Dic-
tio-
nary
ob-
ject
rep-
re-
sent
ing
the
node
suf-
fi-
cien
to
at-
tach
a

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected by cinder drivers. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)



Raises

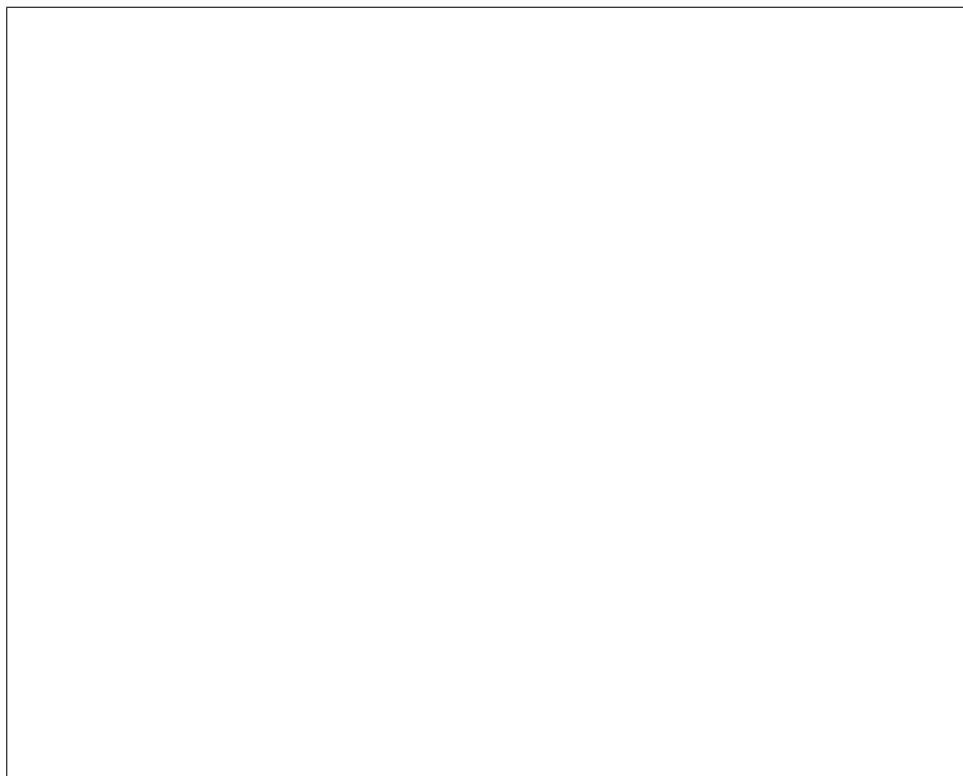
StorageError
StorageError
If storage age subsystem exception is raised

Returns

List

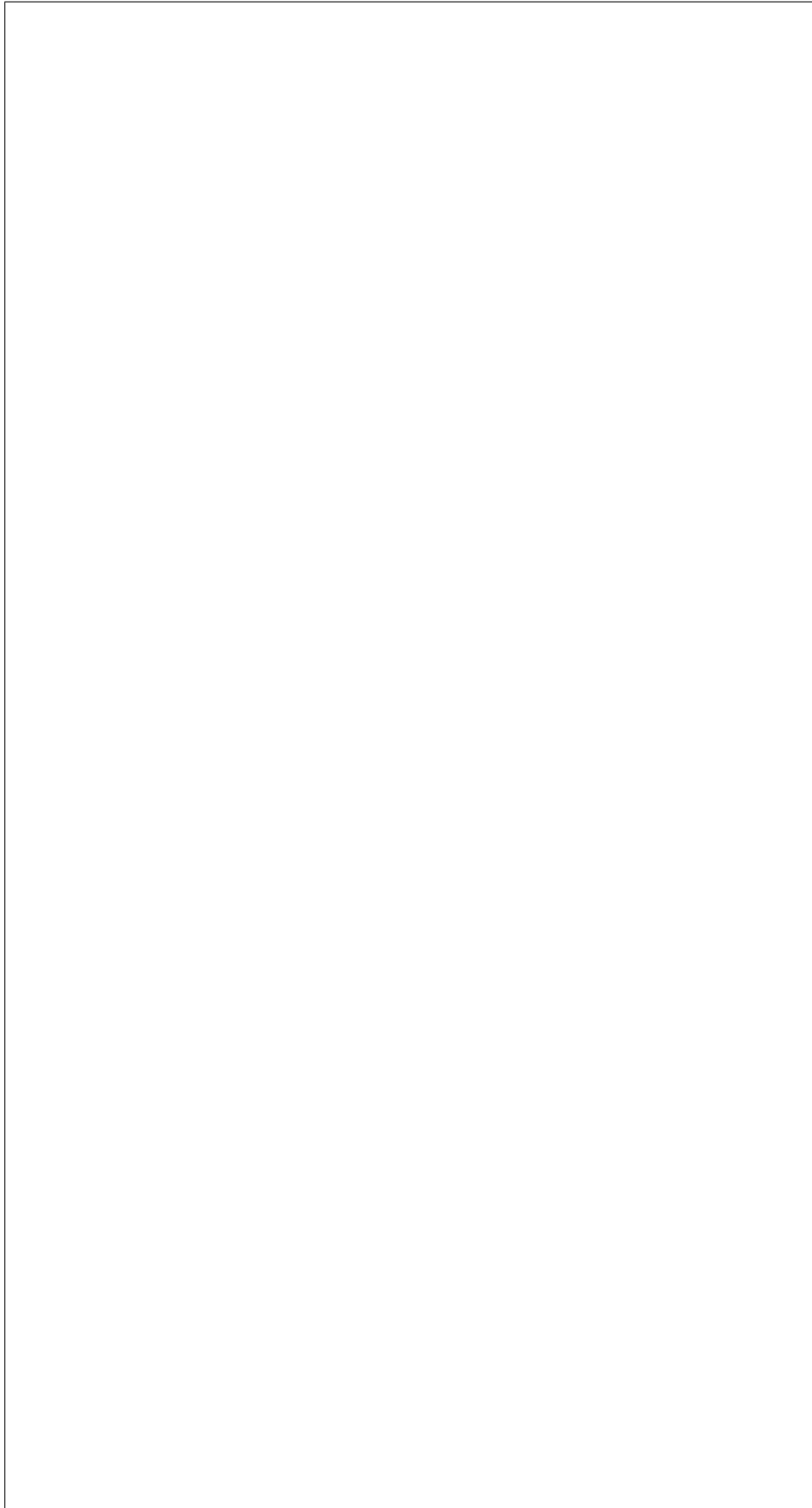
of
con-
nect
vol-
ume
in-
clud
ing
vol-
ume
that
were
al-
read
con-
nect
to
de-
sire

nodes. The returned list can be relatively consistent depending on the end storage driver that the volume is configured for, however the `driver_volume_type` key should not be relied upon as it is a free-form value returned by the driver. The accompanying `data` key contains the actual target details which will indicate either target WWNs and a LUN or a target portal and IQN. It also always contains volume ID in cinder and ironic. Except for these two IDs, each driver may return somewhat different data although the same keys are used if the target is FC or iSCSI, so any logic should be based upon the returned contents. For already attached volumes, the structure contains `already_attached: True` key-value pair. In such case, connection info for the node is already in the database, data structure contains only basic info of volume ID in cinder and ironic, so any logic based on that should retrieve it from the database. Example:



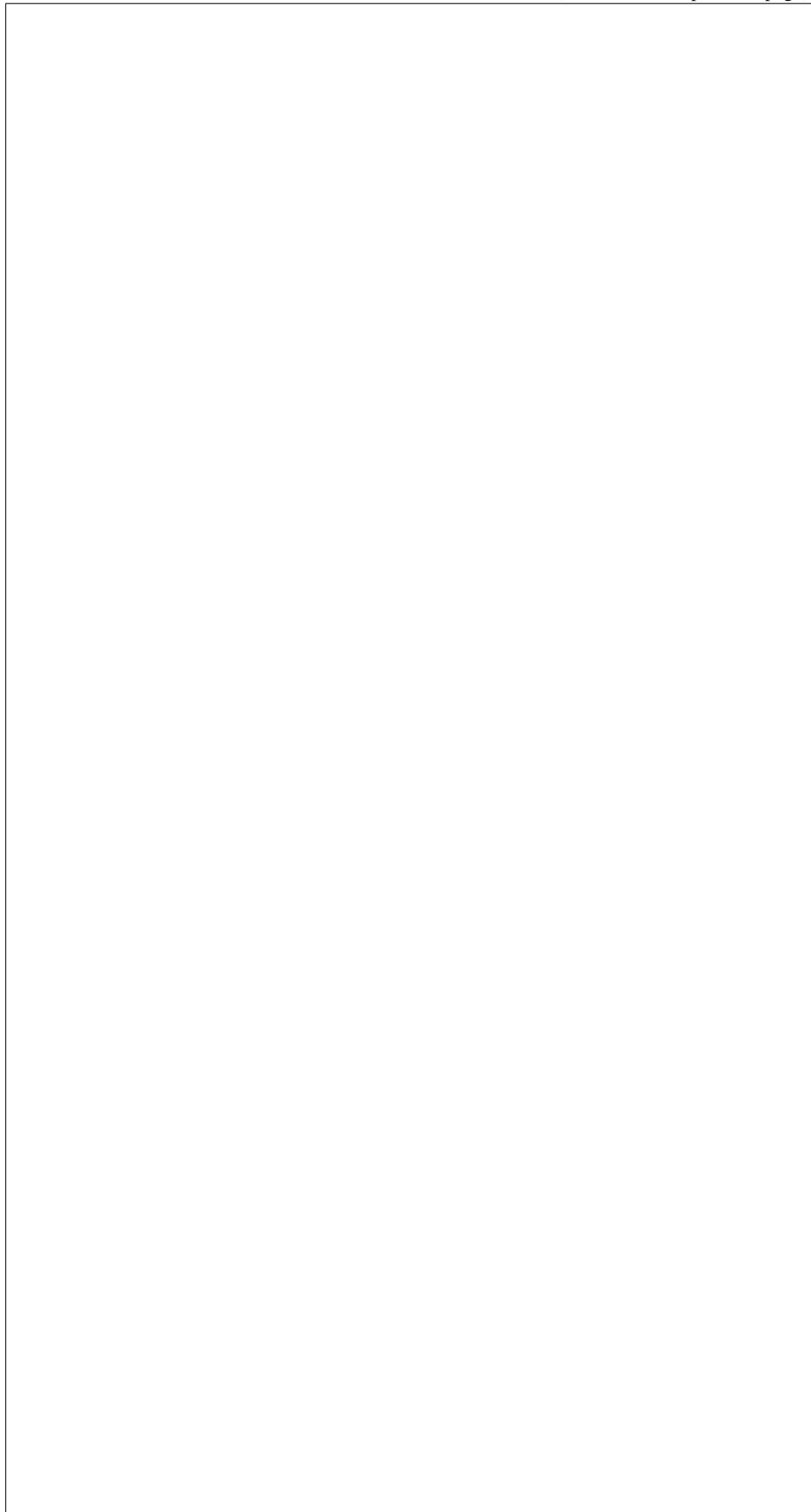
(continues on next page)

(continued from previous page)



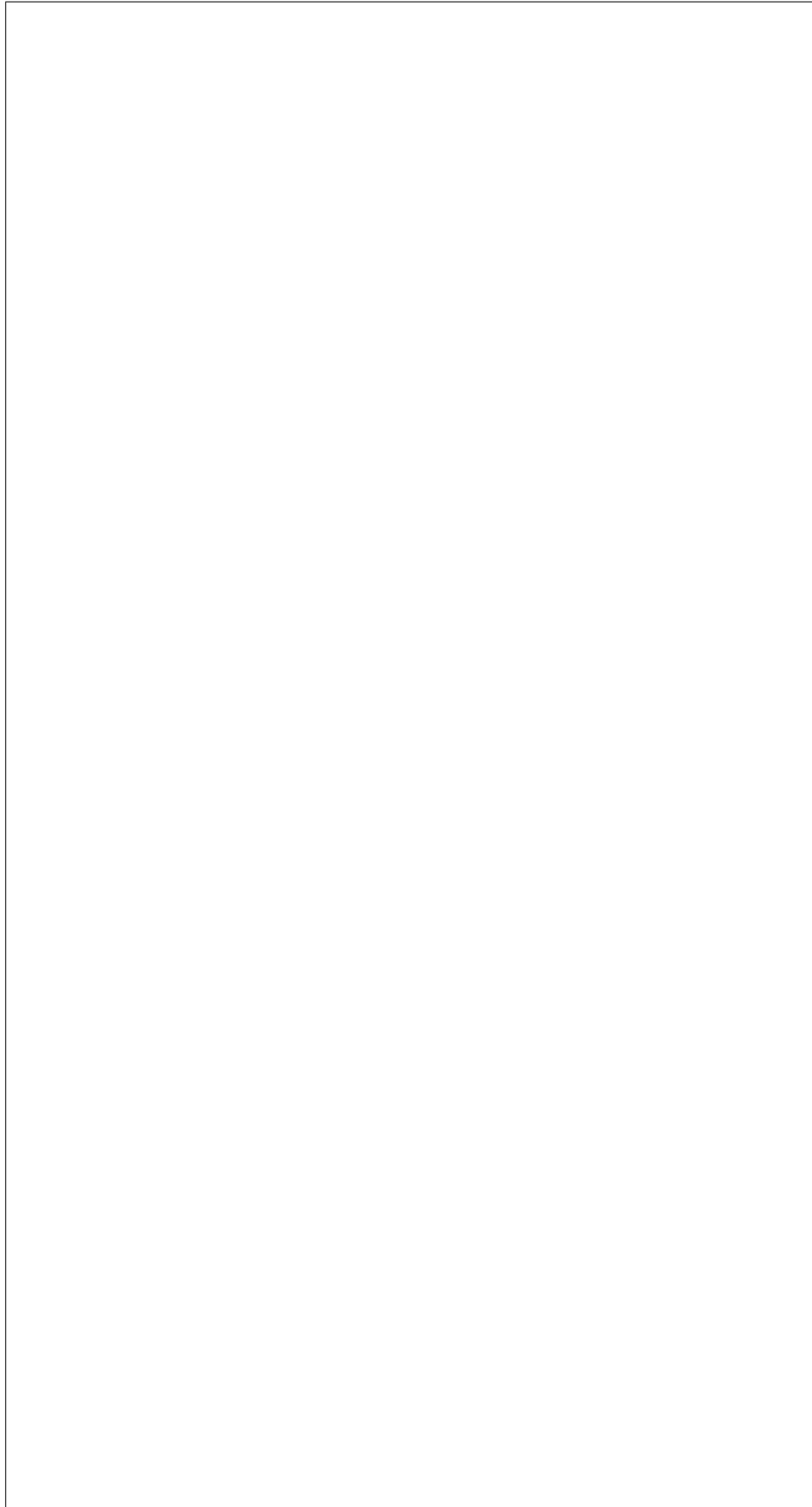
(continues on next page)

(continued from previous page)



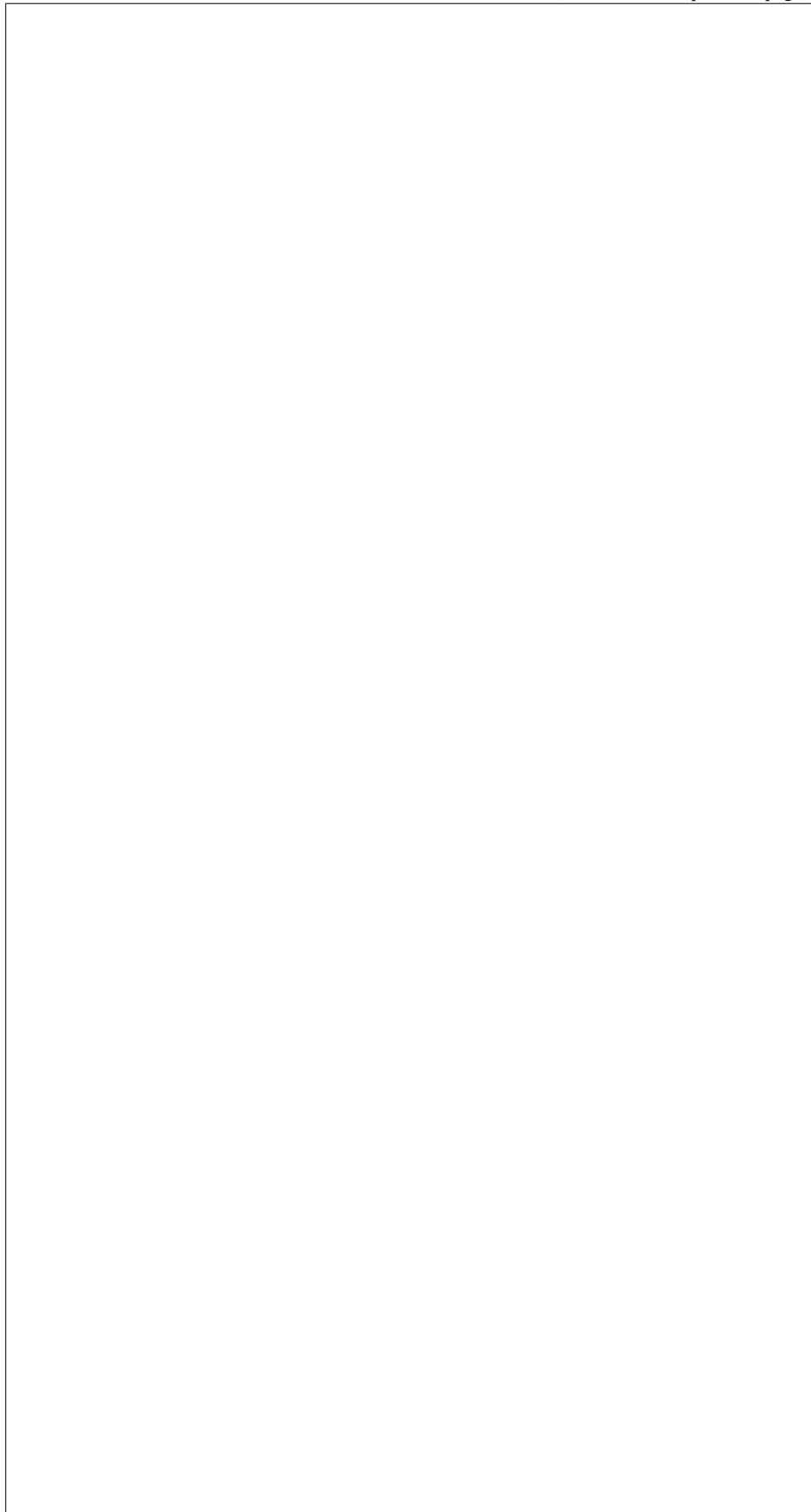
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



ironic.c

De-
tach
a
list
of
vol-
ume
from
a
pro-
vide
con-
nec-
tor
de-
tail.

Enu-
mer-
ates
thro
a
pro-
vide
list
of
vol-
ume
and
is-
sues
de-
tach
men
re-
ques
uti-

lizing the connector information that describes the node.

Paramet

- **task**
The
Task
ager

task
rep-
re-
sent
ing
the
re-
ques

- **vol**
The
list
of
vol-
ume
id
val-
ues
to
de-
tach

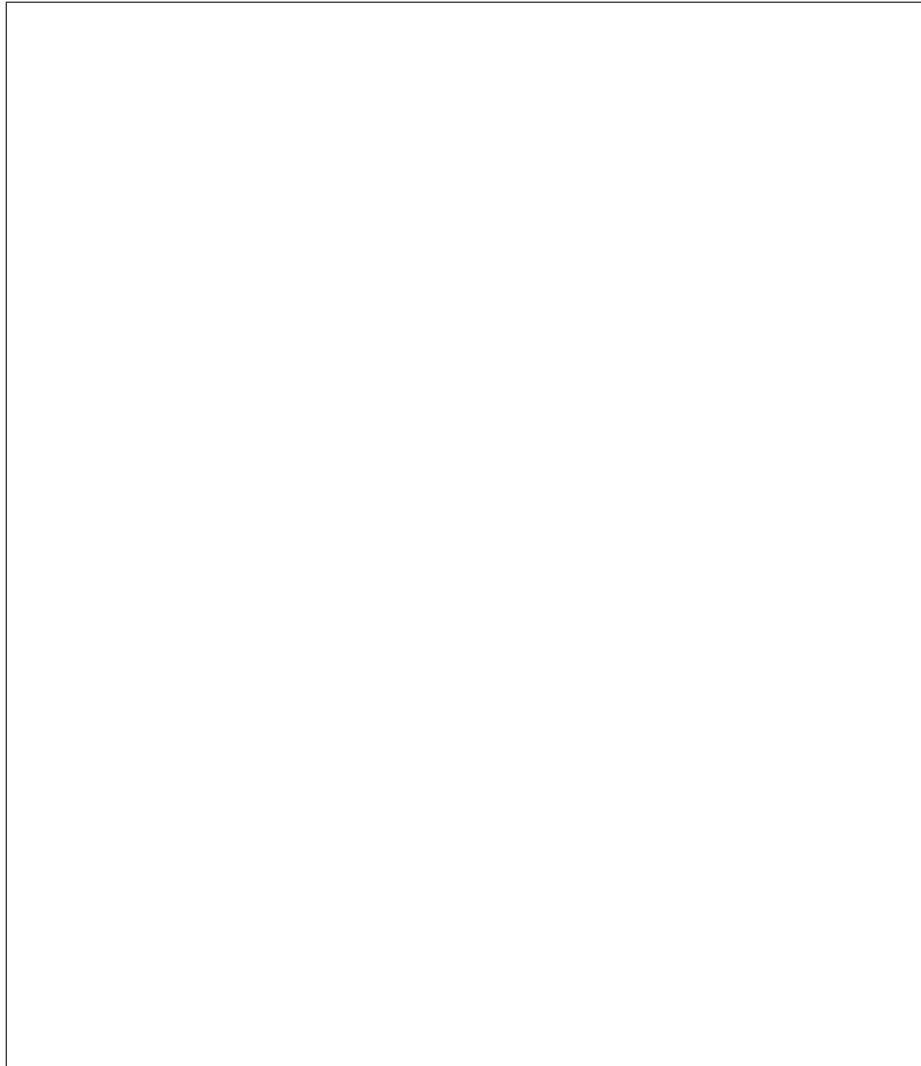
- **conu**
Dic-
tio-
nary
ob-
ject
rep-
re-
sent
ing
the
node
suf-
fi-
cien
to
at-
tach
a

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)



- **all**
Boo
valu
gov-
ern-
ing
if
er-
rors
that
are
re-
turn
are
treat
as
warn
ings
in-

stead of exceptions. Default False.

Raises

StorageError

ironic.c

Get a Cinder client connection.

Parameters

connection: request context, instance of ironic

Returns

A Cinder client

ironic.c

Check if a volume is attached to the supplied node.

Parameters

•

node
The
ob-
ject
rep-
re-
sent
ing
the
node

- **volume**
The
ob-
ject
rep-
re-
sent
ing
the
vol-
ume
from
cin-
der.

Returns

Boo-
in-
di-
cat-
ing
if
the
vol-
ume
is
at-
tach
Re-
turn
True
if
cin-
der
show

the volume as presently attached, otherwise returns False.

ironic.c

Che
if

a
vol-
ume
is
avai
able
for
a
con-
nec-
tion.

Paramet

vol
The
ob-
ject
rep-
re-
sent
ing
the
vol-
ume

Returns

Boo
if
vol-
ume
is
avai
able

ironic.common.components module

Map
ping
of
com
mon
hard
ware
com
po-
nent
of
a
com
pute
sys-

tem.

ironic.c

Cha
sis
en-
clos
ing
one
or
morr
hard
ware
com
po-
nent

ironic.c

Stor
age
drive

ironic.c

Net-
worl
in-
ter-
face

ironic.c

Pow
sup-
ply
unit

ironic.c

Com
put-
ing
sys-
tem

`ironic.common.config` module

`ironic.c`

`ironic.common.context` module

`class ir`

Base
oslo
con
Req
Ex-
tend
se-
cu-
rity
con-
texts
from
the
oslo
li-
brar

`ensure_t`

En-
sure
threa
ing
con-
tains
con-
text
For
asyn
task
the
con-
text
of
lo-
cal
threa
is
miss
ing.

and this is useful to log the request_id in log messages.

logged in user on which it applies policy enforcement. This dictionary defines a standard list of attributes that should be available for enforcement across services.

Set
it
with
re-
ques-
con-
text

to_polic

A
dic-
tio-
nary
of
con-
text
at-
tribu-
to
en-
forc-
pol-
icy
with

oslo
en-
forc-
men
re-
quir
a
dic-
tio-
nary
of
at-
tribu-
rep-
re-
sent
ing
the
cur-
rent

It
is
ex-

pect
that
ser-
vice
will
of-
ten
have
to
over
ride
this
meth
with
ei-
ther
dep-

recated values or additional attributes used by that service specific policy.

ironic.c

Cre-
ate
an
ad-
min-
is-
tra-
tor
con-
text.

ironic.common.dhcp_factory module

class ir

Base
obj

clean_db

Clea
up
the
DHCP
BOC
op-
tions
for
this
node

Parameter

task

A

Task

ager

in-

stan

property

update_c

Send

or

up-

date

the

DHC

BOC

op-

tions

for

this

node

Parameter

-

task

A

Task

ager

in-

stan

-

dhc

this

will

be

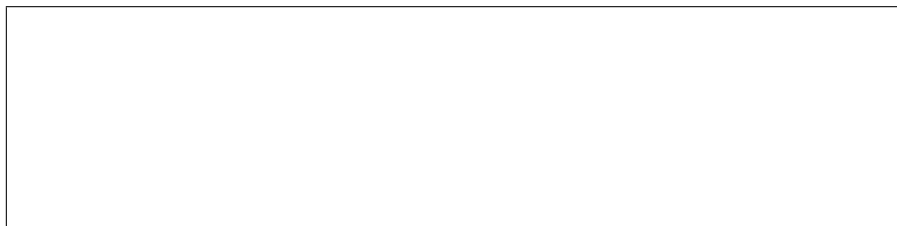
a

list

of

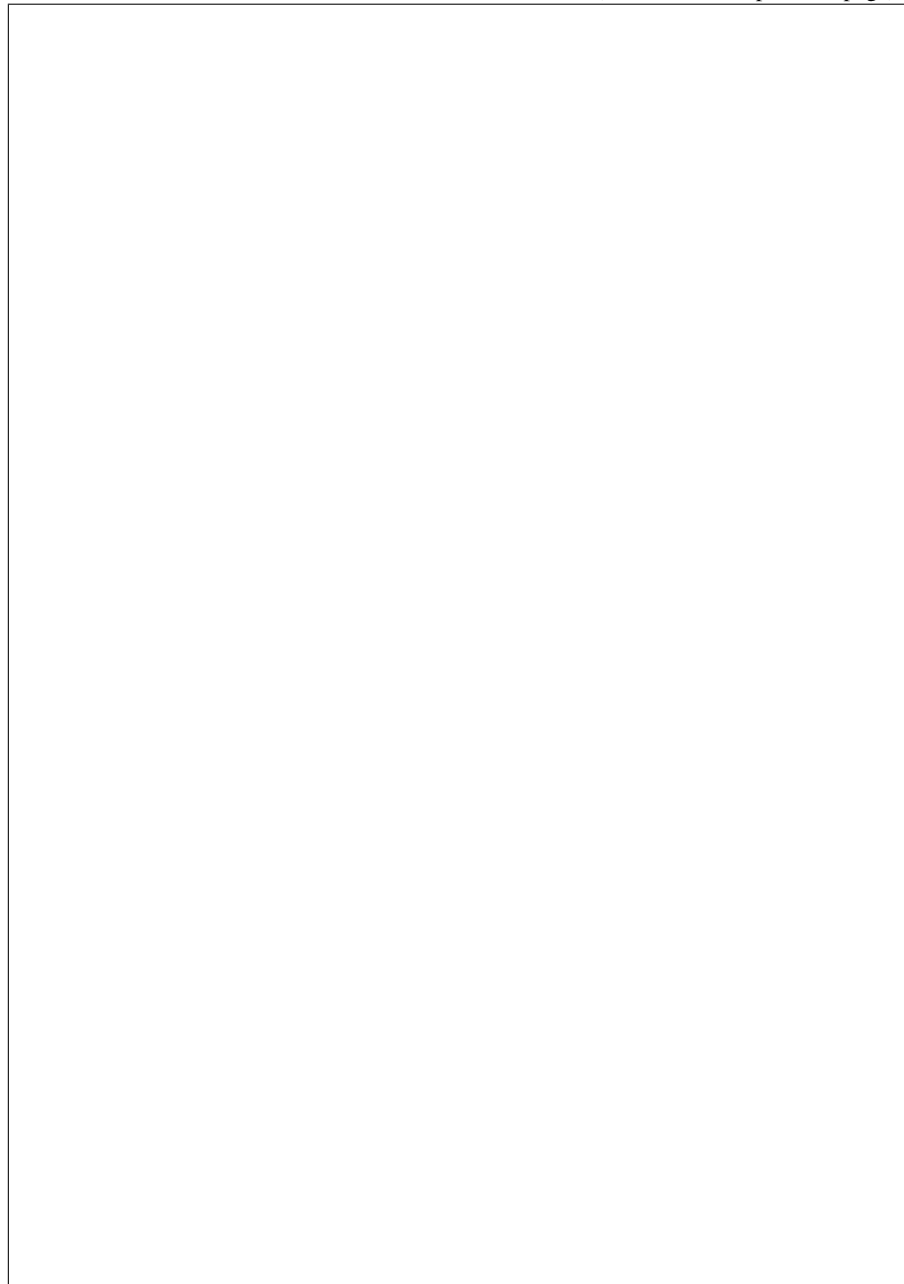
dicts

e.g.



(continues on next page)

(continued from previous page)



- **por**
A
dict
with
keys
port
and
port
grou
and
dicts
as
val-

ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port
ob-

ironic.common.driver_factory module

jects

class ir

Base
obj

Dis-
cove
load
and
man
age
the
driv
avai
able

This
is
sub-
class
to
load
both
main
driv
and
ex-
tra
in-
ter-
face

get_dri

items()

It-
er-
a-
tor
over
pairs
(nam
in-
stan

property

The

list
of
driv
nam
avai
able

class `ir`
Base
iro
com
dri
Bas

class `ir`
Base
iro
com
dri
Bas

class `ir`
Base
iro
com
dri
Bas

`ironic.c`

Get
all
in-
ter-
face
for
all
in-
ter-
face
type

Returns
Dic-
tio-
nary
map-
ping
in-
ter-
face
type
to
dic-

to interface object.

interfaces to it. They come from separate driver factories and are configurable via the database.

tio-
nary
map
ping
in-
ter-
face
nam

ironic.c

Buil
a
com
pos-
able
driv
for
a
give
task

Star
with
a
Bare
ob-
ject,
and
at-
tach
im-
ple-
men
ta-
tions
of
the
var-
i-
ous
driv

Paramet
task
The
task
con-
tain-
ing
the

node
to
build
a
drive
for.

Returns

A
drive
ob-
ject
for
the
task

Raises

Drive
Not-
Found
if
node
could
not
be
found
in
the
iron
nam
space

Raises

In-
ter-
face
Found
nEn
try-
point
if
some
node
in-
ter-
face
are
set
to
in-
valid
or

unsupported values.

ible with it with the hardware type.

Raises

In-
com
pat-
i-
bleI
ter-
face
the
re-
ques
im-
ple-
men
ta-
tion
is
not
com
pat-

`ironic.c`

En-
sure
that
node
in-
ter-
face
(e.g.
for
cre-
ation
or
up-
dat-
ing)
are
valid

Up-
date
(but
does
save
to
the
data
hard
ware

they are not provided.

instance is built for a node.

in-
ter-
face
with
cal-
cu-
latec
de-
fault
if

This
func-
tion
is
run
on
node
up-
dat-
ing
and
cre-
ation
as
well
as
each
time
a
drive

Paramet

- **node**
node
ob-
ject
to
check
and
po-
ten-
tially
up-
date
- **hw_**

hard
ware
type
in-
stan-
ob-
ject;
will
be
de-
tecte
from
node
if
miss
ing

Returns

True
if
any
char
were
mad
to
the
node
oth-
er-
wise
Fals

Raises

In-
ter-
face
Four
nEn
try-
poin
on
val-
i-
da-
tion
fail-
ure

Raises

No-
Vali
De-
fault

is not provided in the configuration

ForI
ter-
face
if
the
de-
fault
valu
can-
not
be
cal-
cu-
latec
and

Raises

Driv
Not-
Four
if
the
node
hard
ware
type
is
not
foun

`ironic.c`

Cal-
cu-
late
and
re-
turn
the
de-
fault
in-
ter-
face
im-
ple-
men
ta-
tion.
Find
the

abled in the configuration.

first
im-
ple-
men-
ta-
tion
that
is
sup-
port
by
the
hard
ware
type
and
is
en-

Parameter

- **hw_**
hard
ware
type
in-
stan-
ce ob-
ject.
- **int**
type
of
the
in-
ter-
face
(e.g.
boot
- **dri**
en-
try-
poin
nam
of
the
hw_

ob-
ject.
Is
used
for
ex-
cep-
tion
mes-
sage

- **node**
the
iden-
ti-
fier
of
a
node
If
spec-
i-
fied,
is
used
for
ex-
cep-
tion
mes-

sage.

Returns
an
en-
try-
poin-
nam-
of
the
cal-
cu-
latec
de-
fault
im-
ple-
men-
ta-
tion.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
en-
try
poin
was
not
foun

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-
face
can
be
foun

`ironic.c`

Get
us-
able
in-
ter-
face
for
a
give
hard
ware
type
For
a
give

faces for each interface type. This is the set of interfaces that are usable for this hardware type.

face names.

hard
ware
type
find
the
in-
ter-
sec-
tion
of
en-
able
and
sup-
port
in-
ter-

Paramet

har
The
hard
ware
type
ob-
ject
to
sear

Returns

a
dict
map
ping
in-
ter-
face
type
to
a
list
of
en-
able
and
sup-
port
in-
ter-

ironic.c

Get
a
hard
ware
type
in-
stan
by
nam

Parameter

hard
the
nam
of
the
hard
ware
type
to
find

Returns

An
in-
stan
of
iron

Raises

Driv
Not-
Four
if
re-
ques
hard
ware
type
can-
not
be
foun

ironic.c

Get
in-
ter-
face
im-
ple-
men

ta-
tion
in-
stan-
For
hard-
ware
type
also
val-
i-
date
com-
pat-
i-
bil-
ity.

Paramet

- **hw_**
a
hard-
ware
type
in-
stan-
- **int**
nam
of
the
in-
ter-
face
type
(e.g.
boot
- **int**
nam
of
the
in-
ter-
face
im-
ple-
men

try point (`ironic.hardware.interfaces.<interface type>`).

ta-
tion
from
an
ap-
pro-
pri-
ate
en-

Returns

in-
stan-
of
the
re-
ques
in-
ter-
face
im-
ple-
men
ta-
tion.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
en-
try
poin
was
not
foun

Raises

In-
com
pat-
i-
bleI
ter-
face
if

plementation is not compatible with it.

hw_
is
a
hard
ware
type
and
the
re-
ques
im-

ironic.c

Get
all
hard
ware
type

Returns

Dic-
tio-
nary
map-
ping
hard
ware
type
nam
to
hard
ware
type
ob-
ject.

ironic.c

Get
all
in-
ter-
face
for
a
give
in-
ter-
face
type

Paramet

int
Strin
type
of
in-
ter-
face
to
fetch
for.

Returns

Dic-
tio-
nary
map-
ping
in-
ter-
face
nam
to
in-
ter-
face
ob-
ject.

ironic.common.exception module

Iron
spe-
cific
ex-
cep-
tions
list.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Not

exceptio

Base
iron
com
exc
Con

exceptio

Base

iron
com
exc
Not.

exceptio

Base
iron
com
exc
Not.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Con.

exceptio

Base
iron
com
exc
Inv.

exceptio

Base
iron
com
exc
Not.

exceptio

Base
Run

property

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Not

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

code = 4

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Not

exceptio

Base

iron
exc
Iron

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Not

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Not.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Not.

exceptio

Base
iron
com
exc
Not.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
IBM

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron

exc
Iro

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
exc
Iro

exceptio

Base
iron
exc
Iro

exceptio

Base
iron
exc
Iro

exceptio

Base
iron
comm
exc
Not.

exceptio

Base
iron
comm
exc
Not.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
comm
exc
Inv.

exceptio

Base
iron
comm
exc
Inv.

exceptio

Base
iron

exc
Iro
exceptio
Base
iro
com
exc
Con
exceptio
Base
iro
exc
Iro
exceptio
Base
iro
com
exc
Not
exceptio
Base
iro
exc
Iro
exceptio
Base
iro
exc
Iro
exceptio
Base
iro
exc
Iro
exceptio
Base

iron
com
exc
Inv

exceptio

Base
iron
exc
Iron

`code = 4`

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base

iron
exc
Iron
exceptio
Base
iron
comm
exc
Inv
exceptio
Base
iron
comm
exc
Inv
exceptio
Base
iron
comm
exc
Cli
property
exceptio
Base
iron
comm
exc
Inv
exceptio
Base
iron
comm
exc
Inv
exceptio
Base
iron
comm
exc
Inv
exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
exc
Iron
exceptio
Base
iron
exc
Iron
exceptio
Base
iron
com
exc
Con
exceptio
Base
iron
com
exc
Inv
exceptio
Base
iron
exc
Iron
exceptio
Base
iron
com
exc
Con
exceptio
Base
iron
exc
Iron
exceptio
Base
iron
com
exc

Temp

code = !

exceptio

Base
iro
com
exc
Temp

exceptio

Base
iro
com
exc
Inv

exceptio

Base
iro
com
exc
Inv

exceptio

Base
iro
com
exc
Not.

exceptio

Base
iro
com
exc
Con

exceptio

Base
iro
com
exc
Inv

exceptio

Base

iron
exc
Iron
exceptio
Base
iron
com
exc
Inv
exceptio
Base
iron
com
exc
Inv
exceptio
Base
iron
com
exc
Inv
exceptio
Base
iron
com
exc
Con.
exceptio
Base
iron
com
exc
Inv
exceptio
Base
iron
com
exc
Not.
exceptio
Base
iron
com

exc

Inv

exceptio

Base

iron

com

exc

HTT

exceptio

Base

iron

exc

Iron

exceptio

Base

iron

com

exc

Not

exceptio

Base

iron

exc

Iron

code = 4

exceptio

Base

iron

exc

Iron

code = 4

exceptio

Base

iron

exc

Iron

code = 4

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Not.

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Inv.

exceptio

Base
iron
exc
Iron

exceptio

Base

iron
com
exc
Con

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Not

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Con

exceptio

Base
iron
com
exc
Inv

exceptio

Base

iron
com
exc
Not

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
com
exc
Inv

exceptio

Base
iron
com
exc
Red

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
com
exc
Dri

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
exc

Ironic

exception

Base

iron

comm

exc

Swi

exception

Base

iron

exc

Ironic

exception

Base

iron

exc

Ironic

code = !

exception

Base

iron

exc

Ironic

exception

Base

iron

comm

exc

cli

property

exception

Base

iron

comm

exc

cli

add_fie

this method will prepend `name` to the hierarchy of names.

Add
a
field
nam
to
con-
cate
nate
the
full
nam
Add
a
field
nam
so
that
the
who
hi-
er-
ar-
chy
is
dis-
play
Suc-
ces-
sive
calls
to

property

exceptio

Base
iro
comm
exc
Inv

exceptio

Base
iron
exc
Iron

exceptio

Base
iron
comm
exc
Con.

exceptio

Base
iron
comm
exc
Con.

exceptio

Base
iron
comm
exc
Inv.

exceptio

Base
iron
comm
exc
Con.

exceptio

Base
iron
comm
exc
Not.

exceptio

Base
iron
comm
exc
Con.

exceptio

Base
iron
comm
exc
Con.

exceptio

Base
ironic
common
exceptions
Common

exceptio

Base
ironic
common
exceptions
Not

exceptio

Base
ironic
exceptions
Iron

ironic.common.faults module

Fault
def-
i-
ni-
tion

ironic.c

Nod
is
mov
to
main
te-
nanc
due
to
fail-
ure
of
a
clea
ing
op-
er-
a-
tion.

`ironic.c`
Node
is
mov
to
main
te-
nanc
due
to
pow
syn-
chro
niza
tion
fail-
ure.

`ironic.c`
Node
is
mov
to
main
te-
nanc
due
to
fail-
ure
of
clea
ing
up
dur-
ing
res-
cue

abort.

`ironic.common.fsm` module

`class ir`
Base
aut
macl
Fin

An
iron
state

mac
class
with
som
iron
spe-
cific
ad-
di-
tions

add_state

Add
a
give
state
to
the
state
ma-
chin

Parameter

- **stable**
Use
this
to
spec
ify
that
this
state
is
a
sta-
ble/p
state
A
state
mus
have
been

previously defined as stable before it can be used as a target

- **target**
The
tar-

target it must have been previously added and specified as stable

get
state
for
state
to
go
to.
Be-
fore
a
state
can
be
used
as
a

Fur-
ther
ar-
gu-
men-
are
in-
ter-
prete
as
for
par-
ent
meth
add.

add_tran

Add
an
al-
lowe
tran-
si-
tion
from
start
-
>
end
for
the
give
even

Parame

- **sta**
start
ing
state
- **end**
end-
ing
state
- **even**
even
that
caus
start
state
to
tran-
si-
tion
to
end
state
- **rep**
re-
plac
ex-
ist-
ing
even
in-
stea
of
rais-
ing
a
Dup
ex-
cep-
tion
whe
the

transition already exists.

initial:

Initial-
tial-
ize
the
FSM

Parame

- **sta**
the
FSM
is
ini-
tial-
ized
to
start
from
this
state

- **tar**
if
spec
i-
fied,
the
FSM
is
ini-
tial-
ized
to
this
tar-
get
state
Oth-
er-
wise

use the default target state

is_stab

Is
the
state
sta-
ble?

Parameter
state
the
state
of
in-
ter-
est

Raises
In-
valid
State
if
the
state
is
in-
valid

Returns
True
if
it
is
a
sta-
ble
state
False
oth-
er-
wise

process_
pro-
cess
the
even

Parameter

- **event**
the
event
to
be
pro-
cess
-

default target state

`ironic.common.hash_ring` module

`ironic.common.i18n` module

target
if
spec
i-
fied,
the
fi-
nal
tar-
get
state
for
the
even
Oth-
er-
wise
use
the

property

class ir

Base
obj

get_ring

classmet

property

oslo
in-
te-
gra-
tion
mod
ule.
See
[http](http://)

```
// docs
oper
org/
oslo
i18n
lates
user
```

ironic.common.image_service module

```
class ir
Base
obj
Pro-
vide
re-
triev
of
disk
im-
ages
```

abstract

Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Paramete

- **ima**
Im-
age
ref-
er-
ence
- **ima**
File

ob-
ject
to
writ
data
to.

Raises
ex-
cep-
tion.

Raises
ex-
cep-
tion.

abstract

Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parameter
image
Im-
age
ref-
er-
ence

Raises
ex-
cep-
tion.

Returns
dic-
tio-
nary
of
im-
age
prop
er-
ties.
It

erties. updated_at attribute is a naive UTC datetime object.

has
three
of
them
size.
up-
date
and
prop

abstract

Val-
i-
date
im-
age
ref-
er-
ence

Parameter

image
Im-
age
ref-
er-
ence

Raises

ex-
cep-
tion.

Returns

In-
for-
ma-
tion
need
to
fur-
ther
op-
er-
ate
with
an
im-
age.

class ir
Base

iron
com
ima
Bas
Pro-
vide
re-
triev
of
disk
im-
ages
avai
able
lo-
cally
on
the
con-
duc-
tor.

download

Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Parameter

- **ima**
Im-
age
ref-
er-
ence
- **ima**
File
ob-
ject

to
writ
data
to.

Raises

ex-
cep-
tion.
if
sour
im-
age
file
does
ex-
ist.

Raises

ex-
cep-
tion.
if
ex-
cep-
tion
were
raise
whil
writ
ing
to
file
or
cre-
at-
ing
hard

link.

show(*ima*

Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parameters

image
Image
reference
reference

Raises

exception.
if
image
file
specified
if
does
exist.

Returns

dictionary
of
image
properties.
It
has
three
of
then
size.
update
and
prop

erties. updated_at attribute is a naive UTC datetime object.

validation

Valid
i-
date
local
im-

age
ref-
er-
ence

Parameter

image
Image
age
ref-
er-
ence

Raises

ex-
cep-
tion.
if
sour-
im-
age
file
does
ex-
ist.

Returns

Path
to
im-
age
file
if
it
ex-
ists.

class ironic

Base
ironic
command
image
Base

Pro-
vide
re-
triev
of
disk
im-
ages
us-
ing

HTT

download

Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Paramete

- **ima**
Im-
age
ref-
er-
ence
- **ima**
File
ob-
ject
to
writ
data
to.

Raises

ex-
cep-
tion.
if
GET
re-
ques
re-
turn
re-
spon
code
not
equa
to

200.

Raises

ex-
cep-
tion.
if:
*
IO-
Er-
ror
hap-
pene
dur-
ing
file
writ
*
GET
re-
ques
faile

show(*ima*

Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Paramete

ima
Im-
age
ref-
er-
ence

Raises

ex-
cep-
tion.
if:
*
HEA
re-
ques

equal to 200; * Content-Length header not found in response to HEAD request.

erties. updated_at attribute is a naive UTC datetime object.

faile
*
HEA
re-
ques
re-
turn
re-
spor
code
not

Returns

dic-
tio-
nary
of
im-
age
prop
er-
ties.
It
has
three
of
them
size,
up-
date
and
prop

validate

Val-
i-
date
HTT
im-
age
ref-
er-
ence

Paramete

- **ima**
Im-

message.

age
ref-
er-
ence

- **sec:**
Spec
ify
if
im-
age_
be-
ing
val-
i-
date
shou
not
be
show
in
ex-
cep-
tion

Raises

ex-
cep-
tion.
if
HEA
re-
ques
faile
or
re-
turn
re-
spor
code
not
equa
to
200.

Returns

Re-
spor
to
HEA
re-

ques
ironic.c
Get
im-
age
ser-
vice
in-
stan
to
dow
load
the
im-
age.

Paramet

- **ima**
Strin
con-
tain-
ing
href
to
get
im-
age
ser-
vice
for.
- **cli**
Glan
clien
to
be
used
for
dow
load
used
only
if
im-
age_
is
Glan

href

- **con**
re-
ques
con-
text,
used
only
if
im-
age_
is
Glan
href

Raises

ex-
cep-
tion.
if
no
im-
age
ser-
vice
can
han-
dle
spec
i-
fied
href

Returns

In-
stan-
of
an
im-
age
ser-
vice
class
that
is
able
to
dow
load
spec
i-

age.

ironic.common.images module

size of the image.

fied
im-

Hand
dling
of
VM
disk
im-
ages

ironic.c

Get
size
of
con-
verte
raw
im-
age.

The
size
of
im-
age
con-
verte
to
raw
for-
mat
can
be
grow
ing
up
to
the
vir-
tual

Paramet

- patl

path
to
the
im-
age
file.

- **est:**
When
to
es-
ti-
mate
the
size
by
scal-
ing
the
orig-
i-
nal
size

Returns

For
*es-
ti-
mate*
re-
turn
the
size
of
the
raw
im-
age
file.
For
*es-
ti-
mate*
re-

turn the size of the original image scaled by the configuration value *raw_image_growth_factor*.

ironic.c

Cre-

this method fetches the kernel and ramdisk, and builds a bootable ISO image that can be used to boot up the baremetal node.

ates
a
boot
ISO
im-
age
for
a
node
Give
the
href:
for
ker-
nel,
ram
root
par-
ti-
tions
UI
and
ker-
nel
cmd
line
ar-
gu-
men

Paramet

- **con**
con-
text
- **outp**
the
ab-
so-
lute
path
of
the
out-
put
ISO

tition image. If not specified, the *esp_image_href* option must be present if UEFI-bootable ISO is desired.

file

- **kernel**
URI
or
glan
uuid
of
the
ker-
nel
to
use

- **ram**
URI
or
glan
uuid
of
the
ram
to
use

- **dep:**
URI
or
glan
UUID
of
the
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-

- **esp:**
URI

the EFI boot loader (e.g. GRUB2) for each hardware architecture to boot. This image will be written onto the ISO image. If not specified, the *deploy_iso_href* option is only required for building UEFI-bootable ISO.

ments of the form $K=V$ or K (optional).

or
gla
UUI
of
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-
tain-
ing

- **ker**
a
strin
con-
tain-
ing
whit
pace
sep-
a-
rate
val-
ues
ker-
nel
cmd
line
ar-
gu-

- **bas**
URI
or
gla
UUI
of
a
to

trieved for to use, instead of building an ISO bootable ramdisk.

image.

be
used
as
an
over
ride
of
wha
shou
be
re-

- **inj**
Map
ping
of
lo-
cal
sour
file
path
to
their
lo-
ca-
tion
on
the
fi-
nal
ISO

Boot_m
the
boot
mod
in
whic
the
de-
ploy
is
to
hap-
pen.

Raises
Im-
age-
Cre-

ation
Fail
if
cre-
at-
ing
boot
ISO
faile

ironic.c

Cre-
ates
an
ESP
im-
age
on
the
spec
i-
fied
file.

Cop
the
pro-
vide
ker-
nel,
ram
and
EFI
sys-
tem
par-
ti-
tion
im-
age
(ESI
to
a
di-

rectory, generates the grub configuration file using kernel parameters and then generates a bootable ISO image for UEFI.

Paramet

- **outp**
the
path
to
the
file
when
the
iso
im-
age
need
to
be
cre-
ated

- **ker**
the
ker-
nel
to
use.

- **ram**
the
ram
to
use.

- **dep**
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-
ti-
tion
im-
age
from
If

not specified, the *esp_image* option is required.

GRUB2) for each hardware architecture to boot. This image will be embedded into the ISO image. If not specified, the *deploy_iso* option is required.

nation of them like $K1=V1,K2$;) to be added as the kernel cmdline.

- **esp**
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-
tain-
ing
the
EFI
boot
load
(e.g.

- **kernel**
a
list
of
string
el-
e-
men-
be-
ing
a
string
like
 $K=V$
or
K
or
com
bi-

- **inject**
Map
ping
of
lo-

image.

command to generate iso.

cal
sour
file
path
to
their
lo-
ca-
tion
on
the
fi-
nal
ISO

Raises

Im-
age-
Cre-
ation
Failure
if
im-
age
cre-
ation
faile
whil
copy
ing
files
or
whil
run-
ning

`ironic.c`

Cre-
ates
an
isoli
im-
age
on
the
spec
i-
fied

ration file using the kernel parameters provided, and then generates a bootable ISO image.

file.
Cop
the
pro-
vide
ker-
nel,
ram
to
a
di-
rec-
tory.
gen-
er-
ates
the
isoli
con-
fig-
u-

Paramet

- **outj**
the
path
to
the
file
whe
the
iso
im-
age
need
to
be
cre-
ated
- **ker**
the
ker-
nel
to
use.
-

nation of them like `K1=V1,K2,)` to be added as the kernel cmdline.

image.

ram
the
ram
to
use.

- **ker**
a
list
of
strin
el-
e-
men
be-
ing
a
strin
like
`K=V`
or
K
or
com
bi-

- **inj**
Map
ping
of
lo-
cal
sour
file
path
to
their
lo-
ca-
tion
on
the
fi-
nal
ISO

Raises
Im-
age-

command to generate iso.

Cre-
ation
Fail
if
im-
age
cre-
ation
faile
whil
copy
ing
files
or
whil
run-
ning

`ironic.c`

Cre-
ates
the
fat
fs
im-
age
on
the
de-
sired
file.

This
meth
copi
the
give
files
to
a
root
di-
rec-
tory
(op-
tion:
writ
the
pa-
ram-

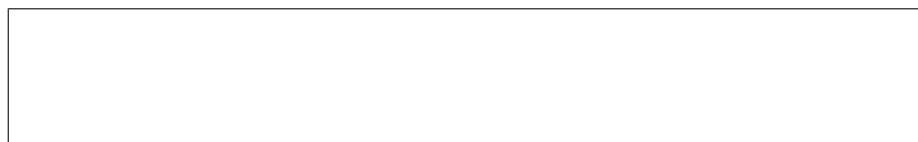
specified to the parameters file within the root directory (optional), and then creates a vfat image of the root directory.

Parameter

- **output**
The path to the file where the vfat image needs to be created.

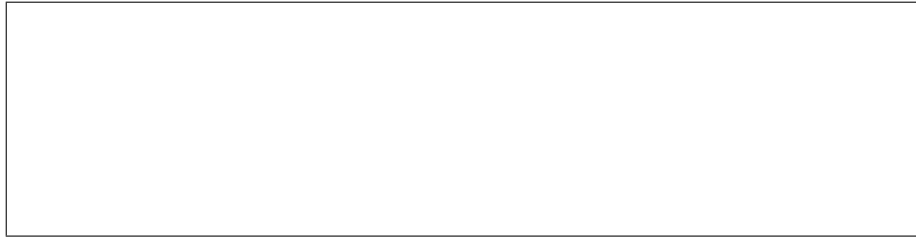
- **file**
A dictionary containing absolute path of file to be copied - > relative

relative path within the vfat image. For example:



(continues on next page)

(continued from previous page)



- **par:**
A dict containing key-value pairs of parameters.

- **par:**
The filename for the parameter file.

- **fs_:**
size of the vfat files in KiB

Raises
Image-Cre-

nipulation activities like creating dirs, mounting, creating filesystem, copying files, etc.

ation
Fail
if
im-
age
cre-
ation
faile
whil
do-
ing
any
of
files
tem
ma-

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

Re-
turn
the
val-
ues
of
sev-
eral
prop
er-
ties
of
an
im-
age

Paramet

- **con**
con-
text
-

ima
href
of
the
im-
age

- **prop**
the
prop
er-
ties
who
val-
ues
are
re-
quir
This
ar-
gu-
men
is
op-
tiona
de-

fault value is all, so if not specified all properties will be returned.

Returns

a
dict
of
the
val-
ues
of
the
prop
er-
ties.
A
prop
erty
not
on
the
glan
meta

data will have a value of None.

ironic.c

ironic.c

Re-
turn
the
tmp
url
for
a
glan
im-
age.

Paramet

- **con**
con-
text
- **ima**
the
UU
of
the
im-
age
in
glan

Returns

the
tmp
url
for
the
glan
im-
age.

ironic.c

ironic.c

ironic.c

Find
out
if
the
im-
age

is
a
par-
ti-
tion
im-
age
or
a
who
disk
im-
age.

Paramet

- **ctx**
an
ad-
min
con-
text
- **ins**
a
node
in-
stan-
info
dict

Returns

True
for
who
disk
im-
ages
and
False
for
par-
ti-
tion
im-
ages
and
Non
on
no

age_source or Error.

ironic.common.indicator_states module

im-

Map
ping
of
the
in-
di-
ca-
tor
LED
state

ironic.c
LED
is
blin
ing

ironic.c
LED
is
off

ironic.c
LED
is
on

ironic.c
LED
state
is
not
know

ironic.common.keystone module

Cent
tral
plac
for
han-
dling
Key
ston

au-
tho-
riza-
tion
and
ser-
vice
look

ironic.c

Loa
adap
from
op-
tion
in
a
con-
fig-
u-
ra-
tion
file
sec-
tion.

The
adap
will
be
pass
di-
rectl
to
key-
ston
Ada
and
will
over
ride
the
val-
ues
load
from

config. Consult keystoneauth1 docs for available adapter options.

Paramet
grou
nam

of
the
con-
fig
sec-
tion
to
load
adap
op-
tions
from

`ironic.c`

Load
auth
plu-
gin
from
op-
tions
in
a
con-
fig-
u-
ra-
tion
file
sec-
tion.

The
auth
will
be
pass
di-
rectl
to
key-
ston
auth
plu-
gin
and
will
over
ride
the
val-

loaded from config. Note that the accepted kwargs will depend on auth plugin type as defined by [group]auth_type option. Consult keystoneauth1 docs for available auth plugins and their options.

Parameter

group
name of the configuration section to load auth plugin options from

ironic.c

Get an endpoint from an adapter

The adapter will be passed directly to keystone Ada and will override the values loaded from

config. Consult keystoneauth1 docs for available adapter options.

Parameter

gro-
nam-
of
the
con-
fig-
sec-
tion
to
load
adap-
op-
tions
from

Raises

Cat-
a-
log-
Not-
Four
if
the
end-
poin
is
not
foun

ironic.c

Cre-
ate
auth
plu-
gin
wrap
ping
both
user
and
ser-
vice
auth

Whe
prop
erly
con-
fig-
ured
and

will not fail if the user token is expired.

isnt serialized yet.

us-
ing
auth
mid-
dle-
ware
re-
ques
with
valid
ser-
vice
auth

Ide-
ally
we
wou
use
the
plu-
gin
pro-
vide
by
auth
mid-
dle-
ware
how
ever
this
plu-
gin

ironic.c

Loa
ses-
sion
ob-
ject
from
op-
tion
in
a
con-
fig-
u-

loaded from config. Consult keystoneauth1 docs for available options.

ra-
tion
file
sec-
tion.
The
ses-
sion
will
be
pass
di-
rectl
to
key-
ston
Ses-
sion
and
will
over
ride
the
val-
ues

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
ses-
sion
op-
tions
from

ironic.c

Wra
key-
ston
func
tions
and

cen-
tral-
izes
ex-
cep-
tion
han-
dling

ironic.common.molds module

ironic.c

Gets
con-
fig-
u-
ra-
tion
mole
from
in-
di-
cate
lo-
ca-
tion.

Paramet

- **task**
A
Task
ager
in-
stan
- **url**
URI
of
the
con-
fig-
u-
ra-
tion
item
to
get.

context.

Returns
JSON
configuration
uration
tion
mole

Raises

- **Ironi**
If
using
Switch
storage
age
and
no
authen
tication
token
found
in
task

- **HTT**
If
failed
to
complete
HTT
re-
ques

ironic.c

Storage
configuration
uration
tion

mole
to
in-
di-
cate
lo-
ca-
tion.

Parameter

- **task**
A
Task
ager
in-
stan

- **name**
URI
of
the
con-
fig-
u-
ra-
tion
item
to
save
to.

- **data**
Con-
tent
of
JSON
data
to
save

Raises

- **Ironi**
If
us-
ing
Swi

context.

`ironic.common.network` module

stor-
age
and
no
au-
then-
ti-
ca-
tion
to-
ken
foun-
in
task

- **HTTP**
If
faile
to
com-
plete
HTTP
re-
ques

`ironic.c`

Get
all
VIF
ids
for
a
node

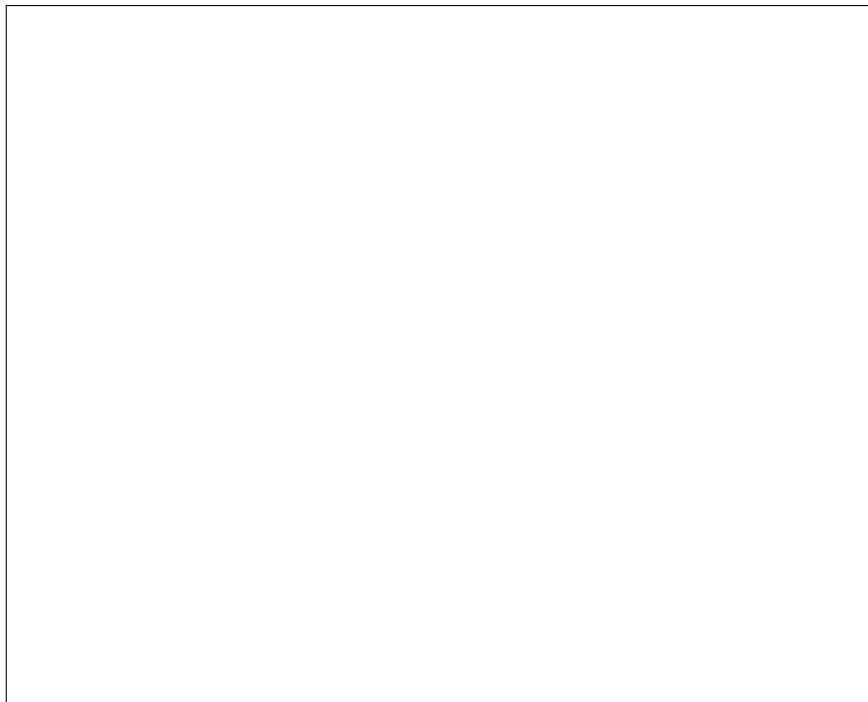
This
func-
tion
does
not
han-
dle
multi-
node
op-
er-
a-
tions

Parameters
task
a
Task
ager
in-
stan

Returns

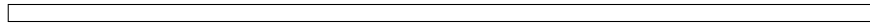
A
dict
of
Node
neu-
tron
port
when
keys
are
port
&
port
group
and
the
val-
ues
are
dict

of UUIDs and their associated VIFs, e.g.



(continues on next page)

(continued from previous page)



ironic.c

Re-
turn
the
set
of
phys
i-
cal
net-
worl
as-
so-
ci-
ated
with
a
port
grou

Paramet

- **task**
a
Task
ager
in-
stan
- **port**
ID
of
the
port
grou
- **exc**
A
Port
ob-
ject
to
ex-
clud
from

physical network, or None.

contain zero or one physical networks.

the
de-
ter-
mi-
na-
tion
of
the
port
group

Returns

The
set
of
phys
i-
cal
net-
work
as-
so-
ci-
ated
with
the
port
group
The
set
will

Raises

Port
group
Phys
net-
Inco
sis-
tent
if
the
port
group
port
are
not
as-
sign
the

ical network.

ical network None is excluded from the set.

sam
phys

ironic.c

Re-
turn
the
set
of
phys
i-
cal
net-
worl
for
a
node

Re-
turn
the
set
of
phys
i-
cal
net-
worl
as-
so-
ci-
ated
with
a
node
port
The
phys

Paramet

tas

a
Task
ager
in-
stan

Returns

A
set

of
phys
i-
cal
net-
worl

ironic.c

Look
a
port
grou
by
ID
on
a
task
ob-
ject.

Paramet

- **task**
a
Task
ager
in-
stan

- **por**
ID
of
the
port
grou

Returns

A
Port
grou
ob-
ject
or
Non

ironic.c

Look
port
by
their

port
grou
ID
on
a
task
ob-
ject.

Paramet

- **task**
a
Task
ager
in-
stan

- **por**
ID
of
the
port
grou

Returns

A
list
of
Port
ob-
jects

`ironic.c`

Re-
mov
all
vif
at-
tach
men
reco
from
a
node

Paramet

task
a
Task

ironic.common.neutron module

ager
in-
stan

class ir
Base
obj

get_clea

get_insp

get_prov

get_reso

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Param

task

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

eters

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

ironic.c

Nam
of
the
neu-
tron
net-
worl
API
phys
i-
cal
net-
worl
pa-
ram-
e-
ter.

ironic.c

Cre-

ate
neu-
tron
port
to
boot
the
ram
Cre-
ate
neu-
tron
port
for
each
pxe_
port
on
task
to
boot
the
ram
If
the
con-
fig
op-
tion
neu-
tron
is
set,
neu-
tron
port
for
non-
pxe-
enab
port
are
also

created these neutron ports will not have any assigned IP addresses.

Paramet

- **tas**
a

Task
ager
in-
stan

- **net**
UUI
of
a
neu-
tron
net-
worl
whe
port
will
be
cre-
ated

- **sec**
List
of
Se-
cu-
rity
Grou
UUI
to
be
used
for
net-
worl

Raises
Net-
worl
Er-
ror

Returns
a
dic-
tio-
nary
in
the
form
{por
neu-

tron.
ironic.c
Re-
triev
a
neu-
tron
clien
con-
nec-
tion.

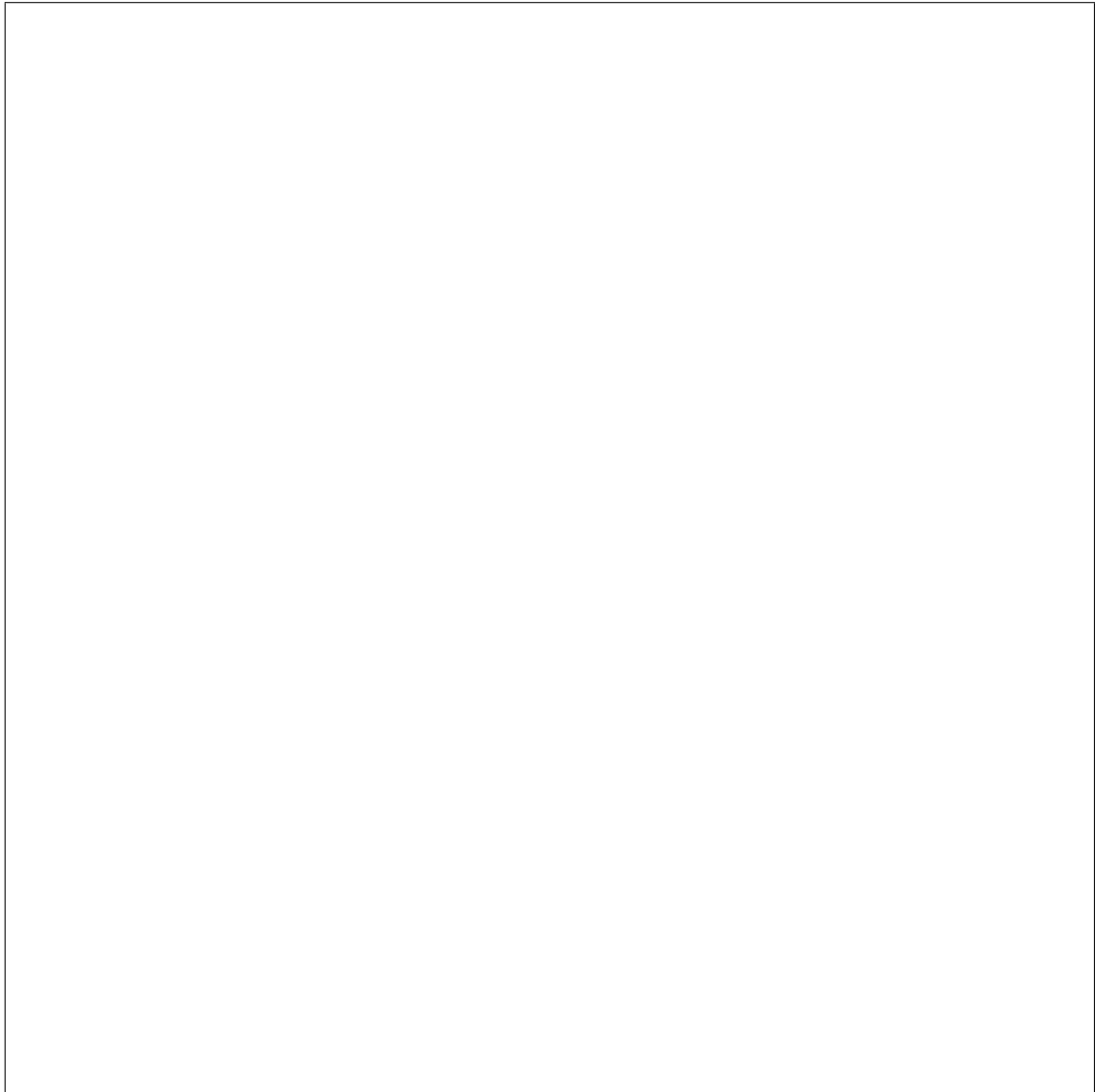
Paramet

- **con**
re-
ques
con-
text,
in-
stan
of
iron
- **autl**
(boo
Whe
True
use
auth
val-
ues
from
conf
pa-
ram-
e-
ters

Returns

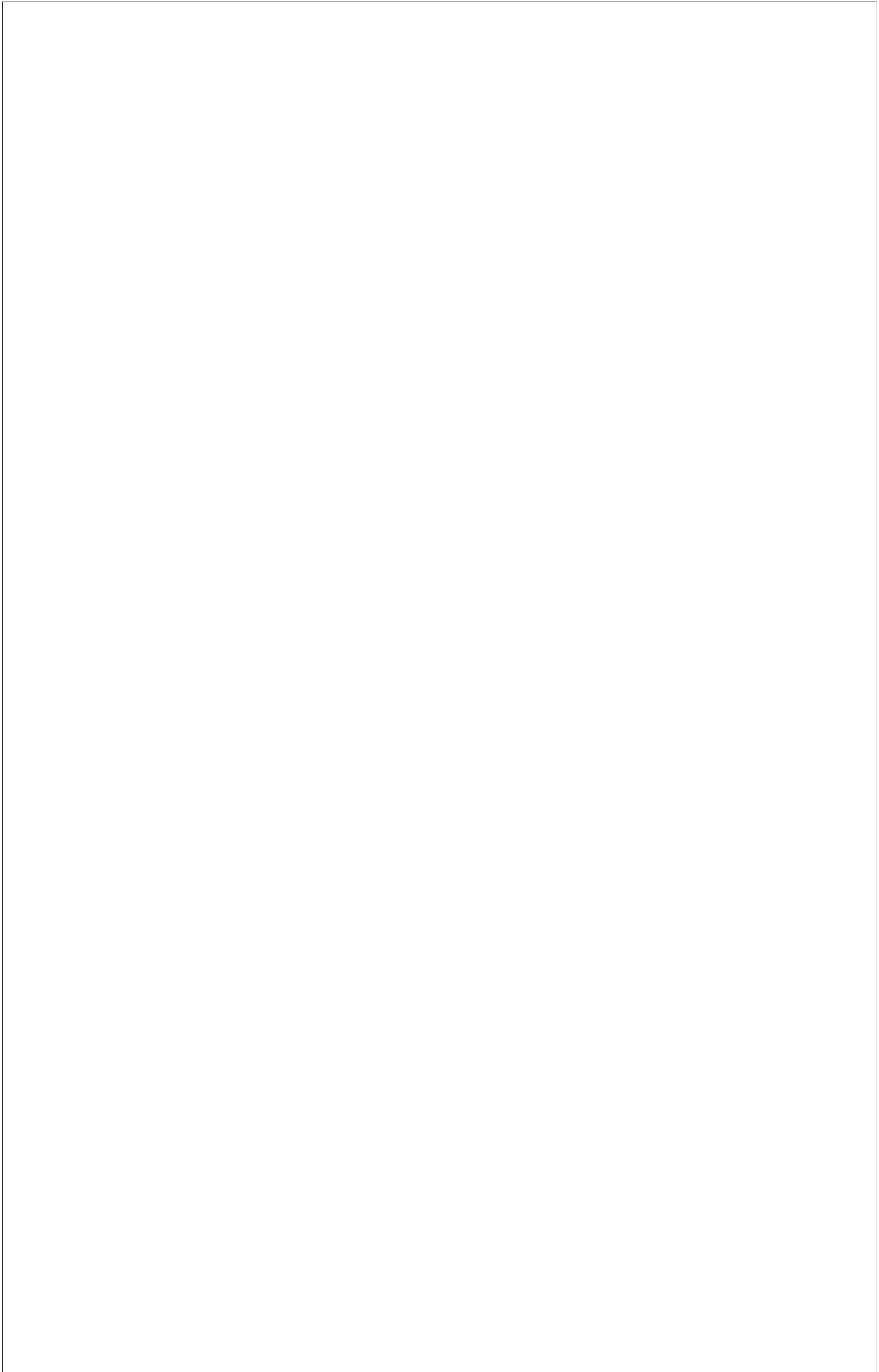
A
neu-
tron
clien
ironic.c
Ex-
tract
the

port
grou
in-
for-
ma-
tion.
The
in-
for-
ma-
tion
is
re-
turn
in
the
form
of:



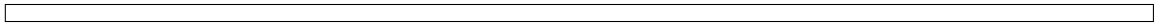
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Parameter

- **task**
a task containing the Node object.
- **port**
Iron port group object to extract data for.

Returns

port group information as a dict

`ironic.c`

Gather Neutron port and network configuration

u-
ra-
tion

Que
Neu
tron
for
port
and
net-
work
con-
fig-
u-
ra-
tion,
re-
turn
wha
ever
is
avai
able

Paramet

- **por**
iron
port
ID.
- **vif**
Neu
tron
port
ID.
- **cli**
Op-
tion
a
Neu
tron
clien
ob-
ject.
- **con**

(irc
com
con
Req
re-
ques
con-
text

Raises

Net-
worl
Er-
ror

Returns

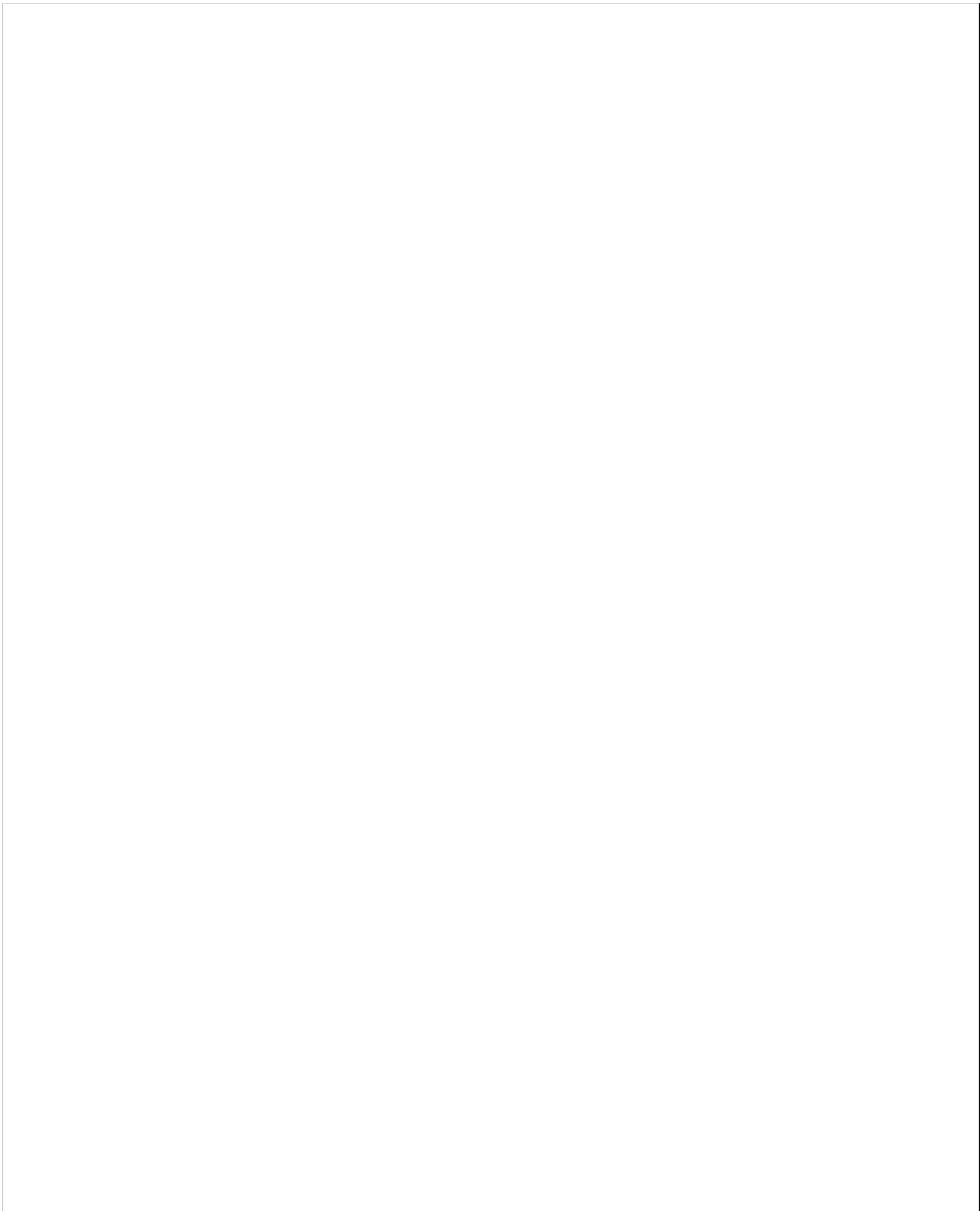
a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
as-
so-
ci-
ated

with this ironic or Neutron port.

ironic.c

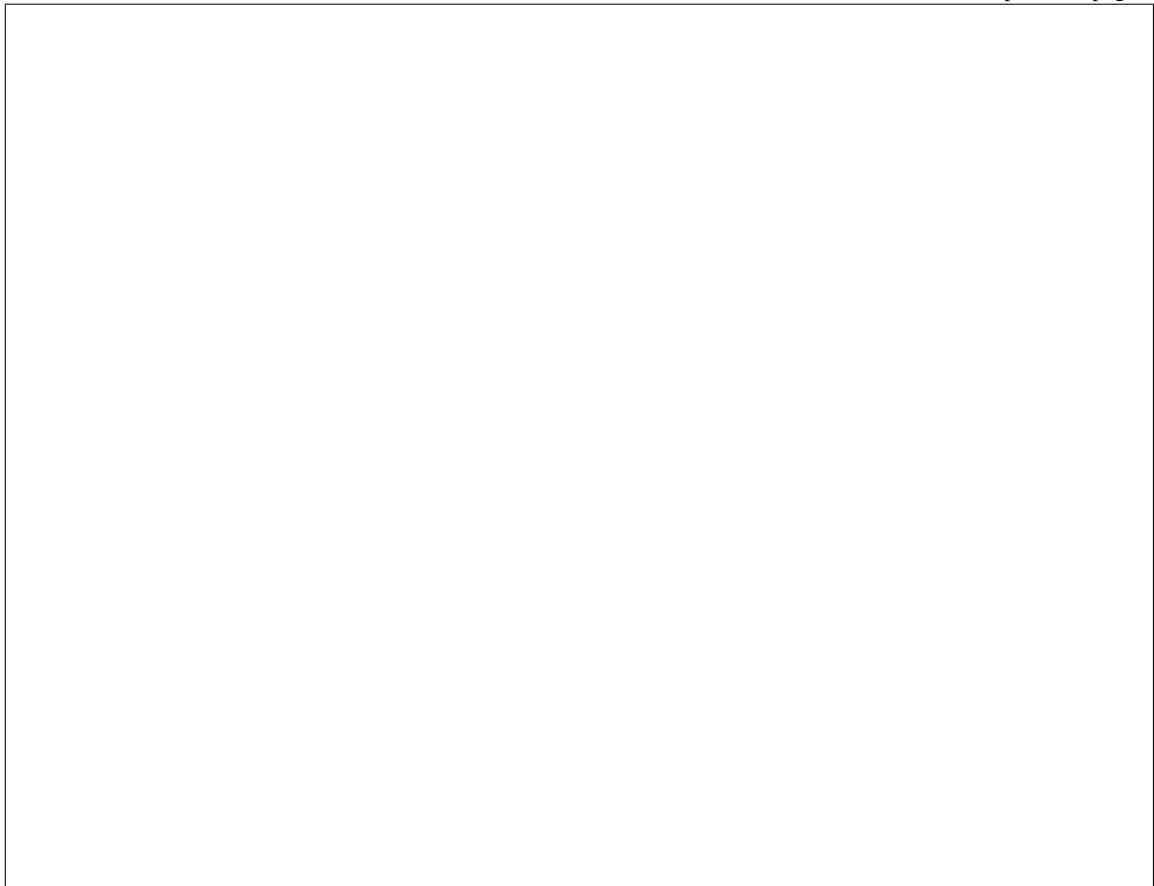
Ex-
tract
the
swit
port
in-
for-
ma-
tion
for
the
node
The

in-
for-
ma-
tion
is
re-
turn
in
the
form
of:



(continues on next page)

(continued from previous page)



Parameter

task

a task containing the Node object.

Returns

port information as a dict

`ironic.c`

Re-

networks associated with the segments in that network.

turn
the
set
of
phys
i-
cal
net-
worl
as-
so-
ci-
ated
with
a
neu-
tron
port
Que
the
net-
worl
to
whic
the
port
is
at-
tach
and
re-
turn
the
set
of
phys
i-
cal

Paramet

- **cli**
A
Neu
tron
clien
ob-
ject.

-

por-
UUI
of
a
Neu
tron
port
to
quer

Returns

A
set
of
phys
i-
cal
net-
worl

Raises

Net-
worl
Er-
ror
if
the
net-
worl
quer
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
for
miss
ing
net-
worl

ironic.c

Che
that
the
port

is
Sma
NIC
port

Paramet

por
an
in-
stan
of
iron
or
port
data
as
dict.

Returns

A
bool
to
in-
di-
cate
port
as
Sma
NIC
port

ironic.c

Dele
the
neu-
tron
port
mat
by
para

Paramet

- **tas**
a
Task
ager
in-
stan
-

par
Dict
of
para
to
fil-
ter
port

Raises

Net-
work
Er-
ror

ironic.c

Dele
the
neu-
tron
port
cre-
ated
for
boot
ing
the
rame

Paramet

- **task**
a
Task
ager
in-
stan
- **net**
UI
of
a
neu-
tron
net-
work
port
will
be
dele

from

Raises

Net-
world
Er-
ror

ironic.c

At-
temp
to
dele
any
port
cre-
ated
by
clea
ing/

Pur-
pose
fully
will
not
raise
any
ex-
cep-
tions
so
er-
ror
han-
dling
can
con-
tinue

Paramet

- **task**
a
Task
ager
in-
stan
- **net**

UUI
of
a
neu-
tron
net-
work

ironic.c

Un-
bind
a
neu-
tron
port

Re-
mov
a
neu-
tron
port
bind
ing
pro-
file
and
host
ID
so
that
it
re-
turn
to
an

unbound state.

Paramet

- **por-**
Neu-
tron
port
ID.
- **cli**
Op-
tion:
a

Neu
tron
clien
ob-
ject.

- **con**
(irc
com
con
Req
re-
ques
con-
text

- **res**
re-
set
mac
ad-
dres

Raises

Net-
worl
Er-
ror

ironic.c

Un-
date
a
neu-
tron
port

Use
neu-
tron
clien
from
conf
clien
to
up-
date
a
neu-
tron
clien

an
un-
bound
state

Parameter

- **con**
re-
ques
con-
text,
in-
stan
of
iron

- **por**
Neu
tron
port
ID.

- **att**
The
at-
tribu
to
up-
date
on
the
port

- **cli**
Op-
tion
Neu
tron
clien

ironic.c

Up-
date
a
port
mac
ad-

dres

Paramet

- **por-**
Neu
tron
port
id.
- **add-**
new
MA
ad-
dres
- **con-**
(irc
com
con
Req
re-
ques
con-
text

Raises

Fail
ToU
date
Port

ironic.c

Che
that
the
give
net-
worl
is
pres

Paramet

- **uui-**
net-
worl
UUI

or
nam

- **net.**
hum
read
net-
worl
type
for
er-
ror
mes
sage

- **con**
(irc
comm
con
Req
re-
ques
con-
text

Returns
net-
worl
UUI

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
uuid
is
emp

Raises
Net-
worl
Er-
ror
on
fail-
ure
to

con-
tact
Neu-
tron

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
for
miss
ing
or
du-
pli-
cate
net-
worl

ironic.c

Che
that
port
con-
tains
enou
in-
for-
ma-
tion
for
de-
ploy
Neu-
tron
net-
worl
in-
ter-
face
re-
quir
that
lo-
cal_
field
is

this port.

filled
be-
fore
we
can
use

Parameter

- **node**
Iron
node
ob-
ject.
- **port**
Iron
port
ob-
ject.

Returns

True
if
port
info
is
valid
False
oth-
er-
wise

`ironic.c`

Wait
for
neu-
tron
agen-
to
be-
com-
tar-
get
state

Parameter

-

cli
A
Neu
tron
clie
ob-
ject.

- **hos**
Age
host

- **tar**
up:
wait
for
up
sta-
tus,
dow
wait
for
dow
sta-
tus

Returns
bool
in-
di-
cate
the
ager
state
mat
para
valu
tar-
get_

Raises
ex-
cep-
tion.
if
tar-
get_
is
not
valid

Raises

tempts.

ex-
cep-
tion.
if
host
sta-
tus
didn
matc
the
re-
quir
sta-
tus
af-
ter
max
retry
at-

ironic.c

Wai
for
port
sta-
tus
to
be
the
de-
sirec
sta-
tus

Paramet

- **cli**
A
Neu
tron
clien
ob-
ject.
- **por**
Neu
tron
port.

-

sta
Port
tar-
get
sta
tus,
can
be
AC-
TIV
DOV
etc.

Returns

bool
in-
di-
cate
that
the
port
sta-
tus
mat
the
re-
quir
valu
pass
by
para
sta-
tus.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
port
does
not
ex-
ist.

Raises

ex-

tempts.

`ironic.common.nova` module

cep-
tion.
if
port
sta-
tus
didn
matc
the
re-
quir
sta-
tus
af-
ter
max
retry
at-

`ironic.c`

Cre-
ates
and
send
pow
state
char
for
the
pro-
vide
serv

Parameter

- **con**
re-
ques
con-
text,
in-
stan
of
iron
-

ser
The
uuid
of
the
node
who
pow
state
char

- **tar**
Tar-
gete
pow
state
char
i.e
POV
or
POV

Returns

A
bool
whic
in-
di-
cate
if
the
pow
up-
date
was
ex-
e-
cute
suc-
cess
fully
(mai

for testing purposes).

ironic.common.policy module

Pol-
icy
En-
gine
For
Iron

ironic.c

A
shor
cut
for
pol-
icy.F

Che
au-
tho-
riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-
den-
tials
and
raise
an
ex-

ception if the rule is not defined. Always returns true if CONF.auth_strategy is not keystone.

ironic.c

A
shor
cut
for
pol-
icy.F

Che
au-
tho-

or False.

riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-
den-
tials
and
re-
turn
True

ironic.c

Con
fig-
u-
ra-
tion
awa
role
pol-
icy
chec
wrap
per.

Che
au-
tho-
riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-
den-
tials
and
re-

or False. Always returns true if CONF.auth_strategy is not keystone.

turn
True

ironic.c

Pro-
vide
ac-
cess
to
the
sin-
gle
in-
stan-
ce
of
Pol-
icy
en-
forc

ironic.c

ironic.c

Syn-
chro-
ni-
tial-
izes
the
pol-
icy
en-
forc

Paramet

- **pol.**
Cus-
tom
pol-
icy
file
to
use,
if
none
is
spec

first instantiation.

i-
fied,
CON
will
be
used

- **rule.**
De-
fault
dic-
tio-
nary
/
Rule
to
use.
It
will
be
con-
sid-
ered
just
in
the

- **def.**
De-
fault
rule
to
use,
CON
will
be
used
if
none
is
spec-
i-
fied.

- **use.**
When
to
load
rules

from
con-
fig
file.

ironic.c

ironic.common.profiler module

ironic.c

Setu
OS-
pro-
filer
no-
ti-
fier
and
en-
able
pro-
fil-
ing.

Paramet

- **name**
name
of
the
ser-
vice
that
will
be
pro-
filed
- **hos**
host
nam
or
host
IP
ad-
dres
that
the

fault host will be set to 0.0.0.0, but specifying host name / address usage is highly recommended.

set in `osprofiler.initializer.init_from_conf`.

ser-
vice
will
be
run-
ning
on.
By
de-

Raises

Typ
in
case
of
in-
valid
con-
nec-
tion
strin
for
a
no-
ti-
fier
back
end,
whic
is

`ironic.c`

Wra
the
OS-
Pro-
filer
trace
dec-
o-
ra-
tor

Wra
the
OS-
Pro-
filer
trace
dec-

unless OSProfiler is present and enabled in the config

o-
ra-
tor
so
that
it
will
not
try
to
patc
the
class

Parameter

- **name**
The name of the action. For example, wsgi, rpc, db, etc..
- **kwargs**
Any other key-words args used by the profiler.

ironic.common.pxe_utils module

class `ironic.common.pxe_utils`
Base class for pxe utils.
`ironic.common.pxe_utils`
`drive`
`mod`
`ima`
`Ima`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

Build the kickstart file for a node.
This method builds the kickstart file for a node based on the template and options for a node.

This method builds the kickstart file for a node based on the template and options for a node. It gives all the requirements for a node.

eters.

The
op-
tions
shou
then
be
pass
to
pxe_
to
cre-
ate
the
ac-
tual
con-
fig
files

Parameter
task

A
Task
ager
ob-
ject

Returns

A
dic-
tio-
nary
of
kick
start
op-
tions
to
be
used
in
the
kick
start
tem-
plate

ironic.c

Buil
the
PXE

con-
fig
op-
tions
for
a
node

This
meth
build
the
PXE
boot
op-
tions
for
a
node
give
all
the
re-
quir
pa-
ram-
e-
ters.

The
op-
tions
shou
then
be
pass
to
pxe_
to
cre-
ate
the
ac-
tual
con-
fig
files

Paramet

- tasl

deployment image kernel and ramdisk info to PXE options.

A
Task
ager
ob-
ject

- **pxe**
a
dict
of
val-
ues
to
set
on
the
con-
fig-
u-
ra-
tion
file
- **ser**
if
True
buil
ser-
vice
mod
pxe
con-
fig
for
netb
ed
user
im-
age
and
skip
addi
- **ipx**
De-
fault
false
bool
to

in-
di-
cate
if
ipxe
is
in
use
by
the
called

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
as
ker-
nel
com
line
ar-
gu-

ments.

Returns

A
dic-
tio-
nary
of
pxe
op-
tions
to
be
used
in
the
pxe
boot
file
tem-

plate
ironic.c

ironic.c
Fetc
the
nec-
es-
sary
ker-
nels
and
ram
for
the
in-
stan

ironic.c
Clea
up
the
TFT
en-
vi-
ron-
men
for
the
task
node

Paramet
task
A
Task
ager
in-
stan

ironic.c
Clea
PXE
en-
vi-
ron-

men
of
all
the
im-
ages
in
im-
ages
Clea
up
the
PXE
en-
vi-
ron-
men
for
the
men
tion
im-
ages
in
im-
ages

Paramet

- **task**
a
Task
ager
ob-
ject
- **ima**
A
dic-
tio-
nary
of
im-
ages
who
keys
are
the
im-

(kernel, ramdisk, etc) and values are a tuple of identifier and absolute path.

age
nam
to
be
clea
up

ironic.c

Ren-
der
the
iPXE
boot
scrip
into
the
HTT
root
di-
rec-
tory

ironic.c

Gen
er-
ate
PXE
con-
fig-
u-
ra-
tion
file
and
MA
ad-
dres
links
for
it.

This
meth
will
gen-
er-
ate
the
PXE
con-

a directory named with the UUID of that node. For each MAC address or DHCP IP address (port) of that node, a symlink for the configuration file will be created under the PXE configuration directory, so regardless of which port boots first theyll get the same PXE configuration. If grub2 bootloader is in use, then its configuration will be created based on DHCP IP address in the form nn.nn.nn.nn.

fig-
u-
ra-
tion
file
for
the
task.
node
un-
der

Parameter

- **task**
A
Task
ager
in-
stan
- **pxe**
A
dic-
tio-
nary
with
the
PXE
con-
fig-
u-
ra-
tion
pa-
ram-
e-
ters.
- **temp**
The
PXE
con-
fig-
u-

cific template will be used.

ra-
tion
tem-
plate
If
no
tem-
plate
is
give
the
node
spe-

ironic.c

Re-
triev
the
DHCP
PXE
boot
op-
tions

Paramet

- **task**
A
Task
ager
in-
stan
- **ipxe**
De-
fault
false
bool
that
sig-
nals
if
iPXE
for-
mat-
ting
shou
be

method for DHCP server configuration.

the node. If `[pxe]ip_version` is set to 6, then this option has no effect as `url_boot` form is required by DHCPv6 standards.

sion. Default to `[pxe]ip_version`. Possible options are integers 4 or 6.

re-
turn
by
the

- **url**
De-
fault
false
bool
to
in-
form
the
meth
if
a
URI
shou
be
re-
turn
to
boot

- **ip_v**
The
IP
ver-
sion
of
op-
tions
to
re-
turn
as
val-
ues
dif-
fer
by
IP
ver-

Returns
Dic-

tions to be set.

tio-
nary
to
be
sent
to
the
net-
work
ing
ser-
vice
de-
scrib
ing
the
DHCP
op-

ironic.c

Gen
er-
ate
ab-
so-
lute
path
to
var-
i-
ous
im-
ages
from
their
nam

This
meth
gen-
er-
ates
ab-
so-
lute
file
sys-
tem
path
on

ous images need to be placed. For example the kickstart template, file and stage2 squashfs.img needs to be placed in the ipxe_root_dir since they will be transferred by anaconda ramdisk over http(s). The generated paths will be added to the image_info dictionary as values.

Parameter

- **node**
the
UI
of
the
node
- **root**
Di-
rec-
tory
in
whic
the
im-
age
mus
be
plac
- **label**
Nam
of
the
im-
age

ironic.c

Gen
er-
ate
http
url
path
to
var-

root. The generated urls will be added to the pxe_options dict and used to render pxe/ipxe configuration templates.

Parameter

- **http**
URI
to
ac-
cess
the
root
of
the
web
serv
- **node**
the
UI
of
the

node

- **label**
Name of the image

ironic.c

Generate the path for TFTP files for deployment or rescue images

This method generates the path for the deployment (or rescue) kernel and deployment (or

rescue) ramdisk.

Parameter

carried out on the node. Supported values are `deploy` and `rescue`. Defaults to `deploy`, indicating deploy operation is being carried out.

- **node**
a
node
ob-
ject

- **mode**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue
op-
er-
a-
tion
be-
ing

- **ipxe**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the
calle
is
us-
ing
iPXE

Returns
a
dic-

cue_kernel, rescue_ramdisk) and values are the absolute paths of them.

tio-
nary
who
keys
are
the
nam
of
the
im-
ages
(de-
ploy
de-
ploy
or
res-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
de-
ploy
or
res-
cue_
is
miss
ing
in
node
driv

ironic.c

Gen
er-
ate
the
path
for
TFT
files
for
in-

dates the node, so caller should already have a non-shared lock on the node.

stand
re-
latec
im-
ages
This
meth
gen-
er-
ates
the
path
for
in-
stan
ker-
nel
and
in-
stan
rame
This
meth
also
up-

Paramet

- **task**
A
Task
ager
in-
stan
con-
tain-
ing
node
and
con-
text.
- **ipx**
De-
fault
false
bool
to

in-
di-
cate
if
ipxe
is
in
use
by
the
called

Returns

a
dic-
tio-
nary
who
keys
are
the
nam
of
the
im-
ages
(ker-
nel,
ram
and
val-
ues

are the absolute paths of them. If its a whole disk image or node is configured for localboot, it returns an empty dictionary.

ironic.c

ironic.c

Get
href
and
tftp
path
for
de-
ploy
or
res-
cue
ker-
nel

and
ram

Paramet

- **node**
UUI
of
the
node
- **drive**
Nod
drive
dict
- **mode**
A
la-
bel
to
in-
di-
cate
whe
path
for
de-
ploy
or
res-
cue
ram
are
be-
- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth

ing requested. Supported values are deploy rescue. Defaults to deploy, indicating deploy paths will be returned.

if
the
called
is
us-
ing
iPXE

Returns

a
dic-
tio-
nary
whose
keys
are
de-
ploy
and
de-
ploy
or
res-
cue_
and
res-
cue_
and

whose values are the absolute paths to them.

Note
drive
shou
be
val-
i-
date
out-
side
of
this
meth

ironic.c

Re-
turn
file
rel-
a-
tive
path
to

CON
Parameter
file
full
file
path
to
be
mad
rel-
a-
tive
path

Returns
The
path
rel-
a-
tive
to
CON

ironic.c
Gen
er-
ate
the
path
for
the
node
PXE
con-
fig-
u-
ra-
tion
file.

Parameter
•
node
the
UUID
of
the
node
•

ipx
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the
calle
is
us-
ing
iPX

Returns

The
path
to
the
node
PXE
con-
fig-
u-
ra-
tion
file.

`ironic.c`

Re-
turn
the
di-
rec-
tory
whe
the
con-
fig
files
and
im-
ages
will
live.

`ironic.c`

Add
trail
ing
slash
(if
need
nec-
es-
sary
for
path
prefi

Returns

CON
en-
sure
to
have
a
trail
ing
slash

`ironic.c`

Iden
tify
vol-
ume
in-
for-
ma-
tion
for
iPXE
tem-
plate
gen-
er-
a-
tion.

`ironic.c`

Re-
turn
true
if
ipxe
is
set.

Parameter

task

A Task object

Returns

boolean
true if the task driver instance is configured or if the task driver instance is the iPX driver

`ironic.c`

Gets the driver specific Node deployment information

This method validates the driver property of

information for this driver to deploy images to, or rescue, the node.

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

the
sup-
plie
node
con-
tain
the
re-
quir

Paramet

- **node**
a
sin-
gle
Nod
- **mode**
La-
bel
in-
di-
cat-
ing
a
de-
plo
or
res-
cue
op-
er-
a-
tion
be-
ing

Returns

A
dict
with
the
driv
val-
ues.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

ironic.c

Pre-
pare
to
boot
ana-
conc
rame
by
gen-
er-
at-
ing
kick
start
file

Paramet

- **task**
a
task
from
Task
ager
- **image**
a
dict
of
val-
ues
of
in-
stan
im-
age
meta
data
to
set

uration file.

on
the
con-
fig-

- **ana**
if
the
boot
is
to
a
ana-
conc
ram
con-
fig-
u-
ra-
tion.

ironic.c

Pre-
pare
the
con-
fig
file
for
PXE
boot

Paramet

- **task**
a
task
from
Task
ager

- **ima**
a
dict
of

uration file.

val-
ues
of
in-
stan-
im-
age
meta-
data
to
set
on
the
con-
fig-

- **iscs**
if
boot
is
from
an
iSCSI
vol-
ume
or
not.

- **ramdisk**
if
the
boot
is
to
a
ram-
con-
fig-
u-
ra-
tion.

- **ipxe**
De-
fault
false
bool
to
in-

di-
cate
if
ipxe
is
in
use
by
the
called

- **ana**
if
the
boot
is
to
a
ana-
conc
rame
con-
fig-
u-
ra-
tion.

Returns
Non

`ironic.c`

Che
if
boot
pa-
ram-
e-
ters
are
valid
for
trust
boot

`ironic.c`

Che
if
the
kick
start
file

is
valid

Parameter

ks_
Contents
of
kick
start
file
to
val-
i-
date

Raises

In-
valid
Kick
start
File

`ironic.c`

Val-
i-
date
the
kick
start
tem-
plate

Parameter

ks_
Path
to
the
kick
start
tem-
plate

Raises

In-
valid
Kick
start
Tem-
plate

ironic.common.raid module

ironic.c

Filter
the
target
raid
con-
fig
base
on
root
vol-
ume
cre-
ation

This
meth
can
be
used
by
any
raid
in-
ter-
face
whic
wan
to
fil-
ter
out
tar-
get
raid

config based on condition whether the root volume will be created or not.

Paramet

- **node**
a
node
ob-
ject
-

root volumes will be filtered out.

else non-root volumes will be filtered out.

cre
A
bool
de-
fault
valu
True
gov-
ern-
ing
if
the
root
vol-
ume
is
re-
turn
else

- **cre**
A
bool
de-
fault
valu
True
gov-
ern-
ing
if
the
non
root
vol-
ume
is
re-
turn

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if

ter skipping root volume and/or non-root volumes.

node
is
miss
ing
or
was
foun
to
be
emp
af-

Returns

It
will
re-
turn
fil-
tere
tar-
get_

`ironic.c`

Get
log-
i-
cal
disk
prop
er-
ties
from
RAI
con-
fig-
u-
ra-
tion
sche

This
meth
read
the
log-
i-
cal
prop
er-
ties
and

that is passed.

may be specified for the logical disk.

their
tex-
tual
de-
scrip-
tion
from
the
sche

Paramet

rai
A
dic-
tio-
nary
whic
is
the
sche
to
be
used
for
get-
ting
prop
er-
ties
that

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
log-
i-
cal
disk
prop
er-
ties
as
keys
and

textual description for them as values.

scheduling purposes (through `properties[capabilities]` and `properties[local_gb]`) and deploying purposes (using `properties[root_device]`).

a
`ironic.c`

Up-
date
the
node
in-
for-
ma-
tion
base
on
the
RAI
con-
fig.

This
meth
up-
date
the
node
in-
for-
ma-
tion
to
mak
use
of
the
con-
fig-
ured
RAI
for

Paramet

- **node**
a
node
ob-
ject
-

is malformed.

rai
The
dic-
tio-
nary
con-
tain-
ing
the
cur-
rent
RAI
con-
fig-
u-
ra-
tion.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
raid
has
more
than
one
root
vol-
ume
or
if
node

ironic.c

Val-
i-
date
the
RAI
con-
fig-
u-
ra-
tion

schema.

pass
us-
ing
JSO
sche

This
meth
val-
i-
date
a
RAI
con-
fig-
u-
ra-
tion
agai
a
RAI
con-
fig-
u-
ra-
tion

Paramet

- **rai**
A
dic-
tio-
nary
con-
tain-
ing
RAI
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
- **rai**

A
dic-
tio-
nary
whic
is
the
sche
to
be
used
for
val-
i-
da-
tion.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
the
RAI
con-
fig-
u-
ra-

tion fails.

ironic.common.release_mappings module

ironic.c

Gets
the
sup-
port
ver-
sion
for
all

ob-
jects

Sup-
port
ver-
sion
are
from
the
RE-
LEA

Paramet

default).

- **rel**
a
list
of
re-
lease
nam
if
emp
ver-
sion
from
all
re-
lease
are
re-
turn
(the

- **obj**
a
list
of
nam
of
ob-
jects
of
in-
ter-
est.
If
emp

jects are returned (the default).

supported versions.

ironic.common.rpc module

ver-
sion
of
all
ob-

Returns

a
dic-
tio-
nary
when
the
key
is
the
ob-
ject
name
and
the
value
is
a
set
of

class ir

Base
oslo
ser
Ser

deserial

De-
se-
ri-
al-
ize
a
dic-
tio-
nary
into
a

re-
ques
con-
text.

Parame

ctx
Re-
ques
con-
text
dic-
tio-
nary

Returns

De-
se-
ri-
al-
ized
form
of
en-
tity

deserial

De-
se-
ri-
al-
ize
some
thing
from
prim
i-
tive
form

Parame

- **ctx**
Re-
ques
con-
text,
in
de-
se-
ri-

al-
ized
form

- **ent.**
Prim
i-
tive
to
be
de-
se-
ri-
al-
ized

Returns

De-
se-
ri-
al-
ized
form
of
en-
tity

seriali

Se-
ri-
al-
ize
a
re-
ques
con-
text
into
a
dic-
tio-
nary

Parame

ctx
Re-
ques
con-
text

Returns

Se-

ri-
al-
ized
form
of
con-
text

serializ

Se-
ri-
al-
ize
som
thing
to
prim
i-
tive
form

Parame

- **ctx**
Re-
ques
con-
text,
in
de-
se-
ri-
al-
ized
form

- **ent**
En-
tity
to
be
se-
ri-
al-
ized

Returns

Se-
ri-
al-

ized
form
of
en-
tity

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

ironic.c

ironic.common.rpc_service module

class ir

Base
oslo
ser
Ser

handle_s

Add
a
sig-
nal
han-
dler
for
SI-
GUS
The
han-
dler

en-
sure
that
the
man-
ager
is
not
dere-
is-
tere-
whe
it
is
shut
dow

start()

Star
a
ser-
vice

stop()

Stop
a
ser-
vice

Parame

gra
in-
di-
cate
whe
to
wait
for
all
thre
to
fin-
ish
or
ter-
mi-
nate
then
in-

stantly

ironic.common.service module

ironic.c

ironic.c

ironic.common.states module

Map
ping
of
bare
meta
node
state

Set-
ting
the
node
power
is
han-
dled
by
the
con-
duc-
tors
pow
syn-
chro
niza
tion
threa
Base

on the power state retrieved from the driver for the node, the state is set to `POWER_ON` or `POWER_OFF`, accordingly. Should this fail, the *power_state* value is left unchanged, and the node is placed into maintenance mode.

The
power
can
also
be
set
man
u-
ally
via

the current state unchanged. The node is NOT placed into maintenance mode in this case.

the
API
A
fail-
ure
to
char
the
state
leav

`ironic.c`
Nod
is
suc-
cess
fully
de-
ploy
and
as-
so-
ci-
ated
with
an
in-
stan

`ironic.c`
Nod
faile
to
com
plete
the
adop
tion
pro-
cess
This
state
is
the
re-
sult-
ing
state
of
a
node

tially due to invalid or incompatible information being defined for the node.

TIVE state to permit designation of nodes as being managed by Ironic, however deployed previously by external means.

that
faile
to
com
plete
adop
tion.
po-
ten-

ironic.c
Nod
is
be-
ing
adop

This
pro-
vi-
sion
state
is
in-
tend
for
use
to
mov
a
node
from
MA-
AGE
ABI
to
AC-

ironic.c
Nod
is
avai
able
for
use
and
sche
ing.
This

state
is
re-
plac
ing
the
NOS
TAT
state
used
prio
to
Kilo

ironic.c
Nod
faile
clea
ing.
This
re-
quir
op-
er-
a-
tor
in-
ter-
ven-
tion
to
re-
solv

ironic.c
Nod
is
be-
ing
au-
to-
mat-
i-
cally
clea
to
pre-
pare
it
for
pro-
vi-

a cleaning step.

sion
ing.

ironic.c
Nod
is
wait
ing
for
a
clea
step
to
be
fin-
ishe

This
will
be
the
node
*pro-
vi-
sion*
whil
the
node
is
wait
ing
for
the
drive
to
fin-
ish

ironic.c
Nod
tear
dow
was
suc-
cess
ful.

In
Junc
tar-
get_
was

target_provision_state.

set
to
this
valu
dur-
ing
node
tear
dow
In
Kilo
this
will
be
a
tran-
si-
tory
valu
of
pro-
vi-
sion
and
neve
rep-
re-
sent
in

ironic.c
failed')
State
in
whic
node
dele
tion
is
al-
lowe

ironic.c
Nod
is
ac-
tivel
be-
ing
torn
dow

alias for ACTIVE.

deployed node should go to ACTIVE status.

ironic.c
Nod
is
suc-
cess
fully
de-
ploy
and
as-
so-
ci-
ated
with
an
in-
stan
This
is
an

ironic.c
Nod
was
suc-
cess
fully
de-
ploy
This
is
main
a
tar-
get
pro-
vi-
sion
state
used
dur-
ing
de-
ploy
men
A
suc-
cess
fully

ironic.c
Nod
de-
ploy
men
faile

ironic.c
Nod
is
read
to
re-
ceiv
a
de-
ploy
re-
ques
or
is
cur-
rentl
be-
ing
de-
ploy

A
node
will
have
its
*pro-
vi-
sion*
set
to
DE-
PLC
ING
brie
be-
fore
it
re-
ceiv
its

initial deploy request. It will also move to this state from DEPLOYWAIT after the callback is triggered and deployment is continued (disk partitioning and image copying).

ironic.c
Nod

deployment.

is
wait
ing
to
be
de-
ploy
This
will
be
the
node
*pro-
vi-
sion*
whil
the
node
is
wait
ing
for
the
drive
to
fin-
ish

ironic.c
Nod
is
en-
rolle
This
state
in-
di-
cate
that
Iron
is
awa
of
a
node
but
is
not
man
ag-

ing
it.

ironic.c

An
er-
ror
oc-
curr
dur-
ing
node
pro-
cess
ing.

The
last_
at-
tribu
of
the
node
de-
tails
shou
con-
tain
an
er-
ror
mes
sage

ironic.c

'clean w

'managea

State
whe
API
look
are
per-
mit-
ted
with
fast
track
en-
able

ironic.c

Nod

in-
spec
tion
faile

ironic.c

Nod
is
un-
der
in-
spec
tion.

This
is
the
pro-
vi-
sion
state
used
when
in-
spec
tion
is
start
A
suc-
cess
fully
in-
spec

node shall transition to MANAGEABLE state. For asynchronous inspection, node shall transition to INSPECTWAIT state.

ironic.c

Nod
is
un-
der
in-
spec
tion.

This
is
the
pro-
vi-
sion
state

cessfully inspected node shall transition to MANAGEABLE state.

used
when
an
asyn
chro
in-
spec
tion
is
in
prog
A
suc-

ironic.c
'cleanin
'rescuin
State
when
API
look
are
nor-
mall
al-
lowe
for
node

ironic.c
Nod
is
in
a
man
age-
able
state

This
state
in-
di-
cate
that
Iron
has
ver-
i-
fied,
at
least

information to manage the hardware. While in this state, the node is not available for provisioning (it must be in the AVAILABLE state for that).

target_*_state fields when there is no target.

once
that
it
had
suf-
fi-
cien

ironic.c
No
state
in-
for-
ma-
tion.

This
state
is
used
with
pow
to
rep-
re-
sent
a
lack
of
know
edge
of
pow
state
and
in

ironic.c
Nod
is
pow
ered
off.

ironic.c
Nod
is
pow
ered
on.

via the REST API.

ironic.c
Nod
is
re-
boot
ing.

ironic.c
Nod
is
to
be
re-
built

This
is
not
used
as
a
state
but
rather
as
a
verb
when
char
ing
the
node
pro-
vi-
sion

ironic.c
Nod
is
in
res-
cue
mod

ironic.c
Nod
res-
cue
faile

ironic.c
Nod
is

rescuing the node.

wait
ing
on
an
ex-
ter-
nal
call-
back
This
will
be
the
node
*pro-
vi-
sion*
whil
the
node
is
wait
ing
for
the
driv
to
fin-
ish

`ironic.c`
Nod
is
in
pro-
cess
of
be-
ing
res-
cuec

`ironic.c`
Nod
is
in
the
pro-
cess
of
soft

pow
off.

ironic.c
Nod
is
re-
boot
ing
grac
fully

ironic.c
'active'
State
that
will
not
tran-
si-
tion
un-
less
re-
ceiv
ing
a
re-
ques

ironic.c
'verify'
State
that
can-
not
be
re-
sum
once
a
con-
duc-
tor
dies

If
a
node
gets
stuck
with
one

when executing task), node will be moved to fail state.

of
thes
state
for
som
rea-
son
(eg.
con-
duc-
tor
goes
dow

ironic.c
Nod
tear
dow
pro-
cess
has
start
This
is
an
alias
for
DEL

ironic.c
Nod
un-
res-
cue
faile

ironic.c
Nod
is
be-
ing
re-
stor
from
res-
cue
mod
(to
ac-
tive
state

ironic.c
'cleanin
wait', '
State
that
can
be
char
with
out
ex-
ter-
nal
re-
ques

ironic.c
'inspect
failed',
Tran
si-
tiona
state
in
whic
we
al-
low
up-
dat-
ing
a
node

ironic.c
'adopt',
'inspect'
'rescue'
Map
ping
of
state
char
even
that
are
PUT
to
the
RES
API

This is a

PUT
/v1/
{tar-
get:
ac-
tive

The dict

{tar-
get
strin
used
by
the
API
in-
ter-
nal
verb

This
pro-
vide
a
ref-
er-
ence
set
of
sup-
port
ac-
tions
and
in
the
fu-
ture
may
be

used to support renaming these actions.

ironic.c
Nod
pow
man
age-
men
cre-
den-
tials
are
be-

ing
ver-
i-
fied.

ironic.c

Use
to
log
whe
en-
ter-
ing
a
state

ironic.c

Use
to
log
whe
a
state
is
ex-
ited.

ironic.common.swift module

class ir

Base
obj

API
for
com
mu-
ni-
cat-
ing
with
Swi

connect:

Un-
der-
ly-
ing
Swi
con-

nec-
tion
ob-
ject.

create_c

Up-
load
a
give
file
to
Swi

Parame

- **con**
The
nam
of
the
con-
taine
for
the
ob-
ject.
- **obj**
The
nam
of
the
ob-
ject
in
Swi
- **fil**
The
file
to
up-
load
as
the
ob-
ject
data

- **obj**
the
head
ers
for
the
ob-
ject
to
pass
to
Swi

Returns
The
Swi
UUI
of
the
ob-
ject

Raises
Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi
fails

delete_c
Dele
the
give
Swi
ob-
ject.

Parame
• **con**
The

nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is
plac

- **obj**
The
nam
of
the
ob-
ject
in
Swi
to
be
dele

Raises
Swi
b-
ject-
Not-
Four
ror,
if
ob-
ject
is
not
foun
in
Swi

Raises
Swi
Op-
er-
a-
tionl
if
op-
er-
a-

tion
with
Swi
fails

get_temp

Re-
turn
the
temp
url
for
the
give
Swi
ob-
ject.

Paramete

- **con**
The
nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is
plac
- **obj**
The
nam
of
the
Swi
ob-
ject.
- **tim**
The
time
out
in

sec-
onds
af-
ter
whic
the
gen-
er-
ated
url
shou
ex-
pire.

Returns

The
temp
url
for
the
ob-
ject.

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi
fails

head_obj

Re-
triev
the
in-
for-
ma-
tion
about
the
give
Swi
ob-

ject.

Parameters

- **con**
The name of the container in which the Switch object is placed

- **obj**
The name of the object in Switch

Returns

The information about the object as returned by the Switch client head call.

Raises

Switch Op-

er-
a-
tion)
if
op-
er-
a-
tion
with
Swi
fails

update_c

Up-
date
the
meta
data
of
a
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is
plac
- **obj**
The
nam
of
the
ob-
ject

in
Swi

- **obj**
the
head
ers
for
the
ob-
ject
to
pass
to
Swi

Raises
Swi
Op-
er-
a-
tion
if
op-
er-
a-
tion
with
Swi
fails

`ironic.c`

ironic.common.utils module

Util-
i-
ties
and
help
func
tions

`ironic.c`

Che
a
di-
rec-
tory

are usable. This should be called from the drivers init function. This function checks that the directory exists and then calls `check_dir_writable` and `check_dir_free_space`. If `directory_to_check` is not provided the default is to use the temp directory.

Parameters

- **dir**
the directory to check
- **req**
amount of space to check for in MiB

Raises

Path
Not-

Four
if
di-
rec-
tory
can
not
be
foun

Raises

Di-
rec-
to-
ryNo
if
user
is
un-
able
to
writ
to
the
di-
rec-
tory

Raises

Ins
if
free
spac
is
<
re-
quir
spac

`ironic.c`

`ironic.c`

Con
ve-
nien
wrap
per
arou
os-
los
ex-
e-

cute
meth

Parameter

- **cmd**
Pass
to
pro-
ces-
su-
til.s.c
- **use**
True
|
Fals
De-
fault
to
Fals
If
set
to
True
ex-
e-
cute
com
man
with
stan

ard locale added to environment variables.

Returns

(std-
out,
stde
from
pro-
cess
ex-
e-
cu-
tion

Raises

Un-
know
nAr-
gu-

men
ror

Raises

Pro-
ces-
sEx-
e-
cu-
tion-
Erro

ironic.c

Che
that
con-
tent
of
the
file
is
the
sam
as
pro-
vide
ref-
er-
ence

Paramet

- **patl**
path
to
file
- **con**
ref-
er-
ence
con-
tent
to
chec
agai
- **hasl**
hash

ing
algo
from
hash
lib
to
use,
de-
fault
is
sha2

Returns

True
if
the
hash
of
ref-
er-
ence
con-
tent
is
the
same
as
the
hash
of
files
con-

tent, False otherwise

`ironic.c`

Gets
a
min
type
of
the
give
file.

`ironic.c`

Re-
turn
an
up-
date
ca-

pabilities. The original capabilities would typically be from a nodes properties[capabilities]. From new_capabilities, any new capabilities are added, and existing capabilities may have their values updated. This updated capabilities string is returned.

pa-
bil-
ity
strin

This
meth
up-
date
the
orig
i-
nal
(or
cur-
rent)
ca-
pa-
bil-
i-
ties
with
the
new
ca-

Paramet

- **cur:**
Cur-
rent
ca-
pa-
bil-
ity
strin
- **new:**
the
dic-
tio-
nary
of
ca-
pa-
bil-
i-

ties
to
be
up-
date

Returns

An
up-
date
ca-
pa-
bil-
ity
strin
with
new

Raises

Val-
ueEr
ror,
if
cur-
rent
is
mal-
form
or
if
new
is
not
a
dic-
tio-
nary

ironic.c

Old
chec
for
valid
log-
i-
cal
node
nam

Re-
tain
for

com
pat-
i-
bil-
ity
with
RES
API
<
1.10

Nominall

- <http://en.wikipedia.org/wiki/Host>
 - <http://tools.ietf.org/html/rfc9>
 - <http://tools.ietf.org/html/rfc1>
- In
prac
tice,
this
chec
has
sev-
eral
shor
com
ings
and
er-

umented in bug #1468508.

rors
that
are
mor
thor
ough
doc-

Paramet

hos
The
host
nam
to
be
val-
i-
date

Returns

True
if
valid
False
if
not.

`ironic.c`

Che
avai
able
sys-
tem
men
ory
and
hold
the
de-
ploy
men
pro-
cess

Eval
u-
ates
the
cur-
rent
sys-

a process by the kernel upon allocation request, and delays the execution until memory has been freed, or until it has timed out.

This is configured using the `[DEFAULT]minimum_memory_wait_time` and the `[DEFAULT]minimum_memory_wait_retries`.

tem
men
ory
avai
able
mea
ing
can
be
al-
lo-
cate
to

This
meth
will
is-
sue
a
sleep
if
the
amo
of
avai
able
men
ory
is
in-
suf-
fi-
cent

Paramet
rai.
De-
fault
Fals
but
if
set
to
true
an
In-
suf-

is raised upon insufficient memory.

fi-
cent
Men
ory
ex-
cep-
tion

Returns

True
if
the
check
has
time
out.
Oth-
er-
wise
Non-
is
re-
turn

Raises

In-
suf-
fi-
cent
Men
ory
if
the
raise
pa-
ram-
e-
ter
is
set
to
True

`ironic.c`

`ironic.c`

Ver-
ify
the
for-
mat

apath ID format: the lower 48-bits are for a MAC address, while the upper 16-bits are implementer-defined.

of
an
Ope
Flow
dat-
a-
p-
ath_
Che
if
a
dat-
a-
p-
ath_
is
valid
and
con-
tains
16
hex-
adec
i-
mal
dig-
its.
Dat-

Paramet

dat:
Ope
Flow
dat-
a-
p-
ath_
to
be
val-
i-
date

Returns

True
if
valid
False
if
not.

ironic.c

De-
ter-
mine
if
a
log-
i-
cal
nam
is
valid
The
log-
i-
cal
nam
may
only
con-
sist
of
RFC
un-
re-
serv
char
ac-
ters,
to
wit:

AL-
PHA
/
DIG
/
-
/
.
/
-
/
~

ironic.c

Che
no_p

is valid.

host names, IP addresses and domain names (with optional :port).

va-
lid-
ity

Che
if
no_
valu
that
will
be
writ
ten
to
en-
vi-
ron-
men
vari-
able
by
iron
pyth
ager

Paramet

no_
the
valu
that
re-
quir
va-
lid-
ity
chec
Ex-
pect
to
be
a
com
sepa
list
of

Returns

True
if
no_
is

valid
Fals
oth-
er-
wise

ironic.c

Mou
a
de-
vice
file
on
spec
i-
fied
lo-
ca-
tion.

Paramet

- **src**
the
path
to
the
sour
file
for
mou
ing
- **des**
the
path
whe
it
need
to
be
mou
- **arg**
a
tu-
ple
con-
tain-

ing
the
ar-
gu-
men
to
be
pass
to
mou
com
man

Raises

pro-
ces-
su-
tils.I
if
it
faile
to
run
the
pro-
cess

`ironic.c`

Pars
the
in-
stan
ca-
pa-
bil-
i-
ties.
One
way
of
hav-
ing
thes
ca-
pa-
bil-
i-
ties
set
is
via

ities are defined in the Flavor extra_spec and passed to Ironic by the Nova Ironic driver.

ity with Juno the Nova Ironic driver is sending it as a string.

Nov
whe
the
ca-
pa-
bil

NOT
Al-
thou
our
API
fully
sup-
port
JSO
field
to
main
tain
the
back
war
com
pat-
i-
bil

**Paramet
nod**

a
sin-
gle
Nod

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-
ties
strin

tionary or is malformed.

tionary.

is
not
a
dic-

Returns

A
dic-
tio-
nary
with
the
ca-
pa-
bil-
i-
ties
if
foun
oth-
er-
wise
an
emp
dic-

ironic.c

Pop
a
valu
from
a
dic-
tio-
nary
field
of
a
node

Paramet

- **node**
Node
ob-
ject.
- **col**

Name
of
the
field
with
the
dic-
tio-
nary

- **file:**
Nest
field
nam

- **def:**
The
de-
fault
valu
to
re-
turn

Returns
The
re-
mov
valu
or
the
de-
fault

`ironic.c`

Ren-
ders
Jinja
tem-
plate
file
with
give
pa-
ram-
e-
ters.

Paramet

- **temp**
full
path
to
the
Jinja
tem-
plate
file
- **par**
dic-
tio-
nary
with
pa-
ram-
e-
ters
to
use
when
ren-
der-
ing
- **is_**
when
tem-
plate
is
file
or
string
with
tem-
plate
it-
self
- **str**
En-
able
strict
tem-
plate
ren-
der-
ing.

De-
fault
is
Fals

Returns

Ren-
dere
tem-
plate

Raises

jinja

ironic.c

ironic.c

Re-
mov
trail
ing
char
ac-
ters
from
a
strin
if
that
does
not
mak
it
emp

Paramet

- **valu**
A
strin
valu
that
will
be
strip
- **cha**
Cha
ac-
ters

to
re-
mov

Returns
Strip
valu

ironic.c

Set
a
valu
in
a
dic-
tio-
nary
field
of
a
node

Paramet

- **node**
Node
ob-
ject.
- **col**
Name
of
the
field
with
the
dic-
tio-
nary
- **field**
Nest
field
name
- **value**
New
valu

ironic.c

ironic.c

Um
a
mou
lo-
ca-
tion.

Paramet

- **loc**
the
path
to
be
un-
mou

- **arg**
a
tu-
ple
con-
tain-
ing
the
ar-
gu-
men-
to
be
pass
to
the
umo
com
man

Raises

pro-
ces-
su-
tils.I
if
it
faile
to

run
the
pro-
cess

ironic.c

ironic.c

Val-
i-
date
an
Ope
Flow
dat-
a-
p-
ath_
and
re-
turn
nor-
mal-
ized
form

Che
whe
the
sup-
plie
Ope
Flow
dat-
a-
p-
ath_
is
for-
mall
cor-
rect
and
nor-
mal-
ize

it to all lower case.

Paramet
dat:
Ope
Flow

dat-
a-
p-
ath_
to
be
val-
i-
date
and
nor-
mal-
ized

Returns

Nor-
mal-
ized
and
val-
i-
date
Ope
Flow
dat-
a-
p-
ath_

Raises

In-
valid
Dat-
a-
p-
athI
If
an
Ope
Flow
dat-
a-
p-
ath_
is
not
valid

`ironic.c`

Val-
i-
date

lower case.

a
MA
ad-
dres
and
re-
turn
nor-
mal-
ized
form

Che
whe
the
sup-
plie
MA
ad-
dres
is
for-
mall
cor-
rect
and
nor-
mal-
ize
it
to
all

Paramet

add:
MA
ad-
dres
to
be
val-
i-
date
and
nor-
mal-
ized

Returns

Nor-
mal-
ized

and
val-
i-
date
MA
ad-
dres

Raises

In-
valid
MA
If
the
MA
ad-
dres
is
not
valid

ironic.c

ironic.c

Val-
i-
date
the
give
port

Paramet

- **por**
TCP
port
- **por**
Nam
of
the
port

Returns

An
in-
te-
ger
port
num

ber.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
port
is
in-
valid

ironic.c

Wra
the
ad-
dres
in
squa
brac
ets
if
its
an
IPv6
ad-
dres

ironic.c

ironic.common.wsgi_service module

class ir

Base
osl
ser
Ser

Pro-
vide
abil-
ity
to
laun
iron

API
from
wsgi
app.

reset()

Re-
set
serv
gree
pool
size
to
de-
fault

Returns

Non

start()

Star
serv
ing
this
ser-
vice
us-
ing
load
con-
fig-
u-
ra-
tion.

Returns

Non

stop()

Stop
serv
ing
this
API

Returns

Non

wait()

Wai
for

the
ser-
vice
to
stop
serv
ing
this
API

Returns
Non

Module contents

`ironic.conductor` package

Submodules

`ironic.conductor.allocations` module

Func
tion-
al-
ity
re-
latec
to
al-
lo-
ca-
tions

`ironic.c`

As-
sign
the
pre-
vi-
ousl
al-
lo-
cate
node
to
the
node
al-
lo-

cation_uuid for a previously allocated node.

ca-
tion.

This
is
not
the
ac-
tual
al-
lo-
ca-
tion
pro-
cess
but
mere
back
fill-
ing
of
al-
lo-

Paramet

- **con**
an
ad-
min
con-
text
- **all**
an
al-
lo-
ca-
tion
ob-
ject
as-
so-
ci-
ated
with
the
node
-

node
An
ID
of
the
node

Raises

Al-
lo-
ca-
tion-
Failure
if
the
node
does
not
match
the
al-
lo-
ca-
tion

Raises

Node
Associated
if
the
node
is
al-
ready
as-
so-
ci-
ated
with
an-
other
in-

stance or allocation.

Raises

In-
stan-
As-
so-
ci-

other node as `instance_uuid`.

ated
if
the
al-
lo-
ca-
tions
UUID
is
al-
read
used
on
an-

Raises

Nod
Not-
Four
if
the
node
with
the
pro-
vide
ID
can-
not
be
foun

`ironic.c`

Pro-
cess
the
al-
lo-
ca-
tion.

This
call
runs
in
a
sep-
a-
rate
threa
on

the allocation and reserves one of them.

a
con-
duc-
tor.
It
finds
suit-
able
nodes
for

This
call
does
not
raise
ex-
cep-
tions
since
its
de-
sign
to
work
asyn-
chro

Parameter

- **con**
an
ad-
min
con-
text
- **all**
an
al-
lo-
ca-
tion
ob-
ject

ironic.c

Ver-
ify

that
al-
lo-
ca-
tion
can
be
re-
mov
for
the
node

Parameter

- **node**
a
node
ob-
ject
- **all**
an
al-
lo-
ca-
tion
ob-
ject
as-
so-
ci-
ated
with
the
node

`ironic.conductor.base_manager` module

Base
con-
duc-
tor
man-
ager
func-
tion-
al-
ity.

class ir

Base
obj

del_host

init_host

Ini-
tial-
ize
the
con-
duc-
tor
host

Parame

adm
the
ad-
min
con-
text
to
pass
to
pe-
ri-
odic
task

Raises

Run
time
ror
whe
con-
duc-
tor
is
al-
read
run-
ning

Raises

NoD
sLoa
whe

no
driv
are
en-
able
on
the
con-
duc-
tor.

Raises

Driv
Not-
Four
if
a
driv
is
en-
able
that
does
not
ex-
ist.

Raises

Driv
Loa
Er-
ror
if
an
en-
able
driv
can-
not
be
load

Raises

Driv
Nam
Con
flict
if
a
clas-
sic
driv
and

have the same name.

a
dy-
nam
driv
are
both
en-
able
and

iter_no

It-
er-
ate
over
node
map
to
this
con-
duc-
tor.

Re-
ques
node
set
from
and
fil-
ters
out
node
that
are
not
map
to
this
con-
duc-
tor.

Yiel
tu-
ples
(nod
driv
con-
duc-
tor_

means yielding (uuid, driver, conductor_group), fields=[foo] means yielding (uuid, driver, conductor_group, foo).

)
when
is
de-
rived
from
field
ar-
gu-
men-
e.g.:
field

Parameters

- **fields:**
list
of
field
to
fetch
in
ad-
di-
tion
to
uuid
driver
and
con-
duc-
tor_
- **kwargs:**
ad-
di-
tiona-
ar-
gu-
men-
to
pass
to
dbap
when
look-
ing
for

node
Returns
gen-
er-
a-
tor
yiel
ing
tu-
ples
of
re-
ques
field

prepare_

Pre-
pare
host
for
ini-
tial-
iza-
tion
Re-
mov
ex-
ist-
ing
data
en-
tries
in-
volv
with
node
lock
ing
for
node
in
a
tran-
si-

tory power state and nodes that are presently locked by the hostname of this conductor.

Un-
der
nor-
mal

op-
er-
a-
tion.
this
is
also
whe
the
ini-
tial
data
con-
nec-
tiv-
ity

is established for the conductors normal operation.

ironic.conductor.cleaning module

Func-
tion-
al-
ity
re-
latec
to
clea
ing.

`ironic.c`

Con-
tinu-
clea
ing
af-
ter
fin-
ish-
ing
an
asyn
clea
step

This
func-
tion
cal-
cu-

lates
whic
step
has
to
run
next
and
pass
con-
trol
into
do_r

Paramet

tasl
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

ironic.c

Do
clea
ing,
start
ing
from
the
spec
i-
fied
clea
step

Paramet

- **tasl**
a
Task
ager
in-
stan

into the list of clean steps in the nodes driver_internal_info[clean_steps]. Is None if there are no steps to execute.

with
an
ex-
clu-
sive
lock

- **step**
The
first
clean
step
in
the
list
to
ex-
e-
cute
This
is
the
in-
dex
(from
0)

- **dis**
When
to
skip
boot
ing
ram
for
clean
ing.

ironic.c

In-
ter-
nal
RPC
meth
to
per-
form
clean

ing
of
a
node

Parameter

- **task**
a
Task
ager
in-
stan-
with
an
ex-
clu-
sive
lock
on
its
node

- **clean**
For
a
man-
ual
clean-
the
list
of
clean-
step-
to
per-
form
Is
Non-
For
au-
to-

ated cleaning (default). For more information, see the `clean_steps` parameter of `ConductorManager.do_node_clean()`.

- **dis**
Whe
to
skip

boot
ing
rame
for
clea
ing.

ironic.c

In-
ter-
nal
meth
to
abor
an
on-
go-
ing
op-
er-
a-
tion.

Paramet

- **task**
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock
- **step**
The
nam
of
the
clea
step

ironic.conductor.deployments module

Functionality related to deploying and undeploying.

`ironic.c`

Continues deployment after finishing an asynchronous deployment step.

This function calculates which step has to run next and passes con-

first run, deploy steps and templates are also validated.

trol
into
do_1
On
the

Paramet

tas
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

ironic.c

Do
de-
ploy
men
start
ing
from
the
spec
i-
fied
de-
ploy
step

Paramet

- **tas**
a
Task
ager
in-
stan
with
an
ex-
clu-
sive

0) into the list of deploy steps in the nodes driver_internal_info[deploy_steps]. Is None if there are no steps to execute.

lock

- **step**
The first de-
ploy step in the list to ex-
e-
cute
This is the in-
dex (from

ironic.c

Pre-
pare the en-
vi-
ron-
men-
and de-
ploy a node

ironic.c

Star-
de-
ploy-
men-
or re-
build-
ing on a node

This function does not check the node suitability for deployment men its left up to the

caller.

Parameter

- **task**
a Task agent instance
- **manager**
a Conductor Manager agent to run task on.
- **config**
a configuration driver if

re-
ques

- **even**
even
to
pro-
cess
de-
ploy
or
re-
buil

- **dep:**
Op-
tiona
de-
ploy
step

`ironic.c`

Val-
i-
date
the
de-
ploy
step
af-
ter
the
rame
learn
about
them

`ironic.c`

Val-
i-
date
that
a
node
is
suit-
able
for
de-
ploy

men

Parameter

- **task**
a
Task
ager
in-
stan
- **event**
even
to
pro-
cess
de-
ploy
or
re-
buil

Raises

Nod
Mai
te-
nan
Nod
Pro-
tecte
In-
valic
State

ironic.conductor.manager module

Con
duct
all
ac-
tiv-
ity
re-
latec
to
bare
meta
de-
ploy

is responsible for performing all actions on bare metal resources (Chassis, Nodes, and Ports). Commands are received via RPCs. The conductor service also performs periodic tasks, eg. to monitor the status of active deployments.

only once, when the ConductorManager service starts. In this way, a single ConductorManager may use multiple drivers, and manage heterogeneous hardware.

men
A
sin-
gle
in-
stan-
of
iron
con
man
Con
is
cre-
ated
with
the
iron
con
pro-
cess
and

Driv
are
load
via
en-
try-
poin
by
the
iron
comm
dri
class
Each
driv
is
in-
stan-
ti-
ated

Wh
mul-
ti-
ple

eratively manage all nodes in the deployment. Nodes are locked by each conductor when performing actions which change the state of that node; these locks are represented by the *ironic.conductor.task_manager.TaskManager* class.

each nodes driver. Rebalancing this ring can trigger various actions by each conductor, such as building or tearing down the TFTP environment for a node, notifying Neutron of a change, etc.

Con
are
run
on
dif-
fer-
ent
host
they
are
all
ac-
tive
and
co-
op-

A
tooz
is
used
to
dis-
tribu
node
acro
the
set
of
ac-
tive
con-
duc-
tors
whic
sup-
port

class *ironic*

Base
ironic
con
bas
Bas
Iron
Con

duc-
tor
man
ager
main
class

RPC_API

add_node

change_r

continue

RPC
meth
to
con-
tinue
clean
ing
a
node

This
is
use-
ful
for
clean
ing
tasks
that
are
asyn
When
they
com
plete
they
call
back
via
RPC

a new worker and lock are set up, and cleaning continues. This can also be used to resume cleaning on `take_over`.

Parame

•

con
an
ad-
min
con-
text.

- **node**
the
id
or
uuid
of
a
node

Raises
In-
valid
State
if
the
node
is
not
in
CLE
WA
state

Raises
NoF
duc-
tor-
Wor
when
there
is
no
free
world
to
start
asyn
task

Raises
Nod
Lock
if
node
is

lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

continue

RPC
meth
to
con-
tinu
de-
ploy
ing
a
node

This
is
use-
ful
for
de-
ploy
ing
task
that
are
asyn
When
they
com
plete
they

RPC, a new worker and lock are set up, and deploying continues. This can also be used to resume deploying on `take_over`.

call
back
via

Parameters

- **con**
an
ad-
min
con-
text.
- **node**
the
ID
or
UUID
of
a
node

Raises

In-
valid
State
if
the
node
is
not
in
DE-
PLC
WA
state

Raises

NoF
duc-
tor-
Wor
when
there
is
no
free
work
to

start
asyn
task

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

create_a

create_p

create_p

destroy_

destroy_

destroy_

destroy_

destroy_

destroy_

do_node

do_node

do_node

do_node

do_node

do_provi

driver_v

get_boot

get_cons

get_driv

get_driv

get_ind:

get_node

get_node

get_rai

get_sup

get_sup

heartbe

inject_r

inspect,

object_a

Per-
form
an
ac-
tion

on
a
Ver-
sion
dOb
ject
in-
stan

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
ac-
tion
- **obj**
The
ob-
ject
in-
stan
on
whic
to
per-
form
the
ac-
tion
- **obj**
The
nam
of
the
ac-
tion
meth
to
call

-

arg
The
po-
si-
tion:
ar-
gu-
men
to
the
ac-
tion
meth

-

kwargs
The
key-
word
ar-
gu-
men
to
the
ac-
tion
meth

Returns

A
tu-
ple
with
the
up-
date
mad
to
the
ob-
ject
and
the
re-
sult
of
the
ac-

tion method

object_L

ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

Per-
form
a
back
port
of
an
ob-
ject
in-
stan-

The
de-
fault
be-
hav-
ior
of
the
base
Ver-
sion
dOb-
ject-
Se-
ri-
al-
izer,
upon
re-
ceiv-

Parame

- **con**
The
con-
text
with
which
to
per-
form
the
back
port

- **obj:**
An
in-
stan-
ce
of
a
Ver-
sion
dOb-
ject
to
be
back-
port-

- **obj:**
A
dict-
ion-
ary
{ob-
j-
nam-
e-
ver-
sion
map-
ping

Returns

The
down-
grad-
ed
in-
stan-
ce
of
ob-
jinst

object_

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb-
ject
class

Parameter

- **context**
The context with which to perform the action

- **object**
The registry name of the object

- **object_name**
The name of the action method to call

- **object_version_map**
A dictionary of {object_name: version} mapping

arg:
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs:**
The
key-
word
ar-
gu-
men
to
the
ac-
tion
meth

Returns

The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)
be
an
in-
stan-
ce
of
the

implementing VersionedObject class.

remove_r

set_boot

set_cons

set_ind

set_targ

target =

update_r

update_p

update_p

update_v

update_v

validate

vendor_p

vif_atta

vif_deta

vif_list

ironic.c

Syn
the
pow
state
for
this
node
in-
cre-
men
ing
the
cour
on
fail-
ure.

When
the
limit
of
power
is
reac
the
node
is
put
into
main
te-
nan
mod
and
the
er-
ror

recorded.

Parameter

- **task**
a
Task
ager
in-
stan
- **count**
num
ber
of
time
this
node
has
pre-
vi-
ousl
faile
a
sync

Raises

Node
Lock
if

un-
able
to
up-
grad
task
lock
to
an
ex-
clu-
sive
one

Returns

Cou
of
faile
at-
temp
On
suc-
cess
the
cour
is
set
to
0.
On
fail-
ure,
the
cour

is incremented by one

ironic.c

ironic.c

Han
dles
pow
state
sync
ex-
ceed
ing
the
max

imum number of retries, change the DB power state to be the actual node power state and place the node in maintenance.

re-tries
When
syn-chroniz-
ing the
power
state
be-tween
a
node
and
the
DB
has
ex-ceeded
the
max

Parameter

- **task**
a
Task
ager
in-
stan-
with
an
ex-
clu-
sive
lock
- **act**
the
ac-
tual
pow-
state
of
the
node

a
pow
state
from
iron

•
exc
the
ex-
cep-
tion
ob-
ject
that
caus
the
sync
pow
state
to
fail,
if
pres

ironic.conductor.notification_utils module

ironic.c

Help
for
con-
duc-
tor
send
ing
a
set
con-
sole
state
no-
ti-
fi-
ca-
tion.

Paramet

•
tas

a
Task
ager
in-
stand

- **act.**
Ac-
tion
strin
to
go
in
the
Ever
Type
Mus
be
ei-
ther
con-
sole,
or
con-
sole.

- **sta**
One
of
iron
END
or
ER-
ROF

ironic.c

Help
for
con-
duc-
tor
send
ing
a
set
pow
state
no-
ti-
fi-

ca-
tion.

Parameter

- **task**
a
Task
ager
in-
stand

- **level**
No-
ti-
fi-
ca-
tion
level
One
of
iron

- **status**
Sta-
tus
to
go
in
the
Ever
Type
One
of
iron
or
ER-
ROF
ER-
ROF
in-
di-

cates that ironic-conductor couldnt retrieve the power state for this node, or that it couldnt set the power state of the node.

- **to_p**
the
pow
state

used instead of the nodes `target_power_state` attribute since the `baremetal.node.power_set.start` notification is sent early, before `target_power_state` is set on the node.

the
con-
duc-
tor
is
at-
temp
ing
to
set
on
the
node
This
is

`ironic.c`

Help
for
con-
duc-
tor
send
ing
a
node
pow
state
cor-
rect
no-
ti-
fi-
ca-
tion.

When
iron
de-
tect
that
the
ac-
tual
pow
state
on
a

ent from the power state on an ironic node (DB), the ironic nodes power state is corrected to be that of the bare metal hardware. A notification is emitted about this after the database is updated to reflect this correction.

Parameter

- **task**
a TaskManager instance
- **from**
the power state of the node before this character was detected

`ironic.c`

Help for conductor sending a set

pro-
vi-
sion
state
no-
ti-
fi-
ca-
tion.

Paramet

- **task**
a
Task
ager
in-
stan

- **lev**
One
of
field

- **sta**
One
of
field

- **pre**
Pre-
vi-
ous
pro-
vi-
sion
state

- **pre**
Pre-
vi-
ous
tar-
get
pro-
vi-
sion
state

ironic.conductor.rpcapi module

- **even**
FSM
even
that
trig-
gere
pro-
vi-
sion
state
char

Clie
side
of
the
con-
duc-
tor
RPC
API

class ir

Base
obj

Clie
side
of
the
con-
duc-
tor
RPC
API

API
ver-
sion
his-
tory

1.0
-

Ini-
tial
ver-
sion

Incl
get_

1.1
-

Add
up-
date
and
start

1.2
-

Add
ven-
dor_

1.3
-

Re-
nam
start
to
char

1.4
-

Add
do_1
and
do_1

1.5
-

Add
val-
i-
date

1.6

-

char
do_r
and
do_r

acce
node
id
in-
stea
of
node
ob-
ject.

1.7

-

Add
topic
pa-
ram-
e-
ter
to
RPC
meth
ods.

1.8

-

Add
char

1.9

-

Add
de-
stroy

1.10

-

Re-
mov
get_

1.11
-

Add
get_
set_

1.12
-

val-
i-
date
do_y
re-
plac
by
sin-
gle

veno
meth

1.13
-

Add
up-
date

1.14
-

Add
driv

1.15
-

Add
re-
buil
pa-
ram-
e-
ter
to
do_1

1.16

-

Add

get_

1.17

-

Add

set_

get_

and

get_

1.18

-

Re-

mov

char

1.19

-

Cha

re-

turn

valu

of

ven-

dor_

and

driv

1.20

-

Add

http

pa-

ram-

e-

ter

to

ven-

dor_

and

driv

1.21

-

Add

get_

and

get_

1.22

-

Add

con-

fig-

driv

pa-

ram-

e-

ter

to

do_1

1.23

-

Add

do_1

1.24

-

Add

in-

spec

metl

1.25

-

Add

de-

stroy

1.26

-

Add
con-
tinue

1.27
-

Con-
vert
con-
tinue
to
cast

1.28
-

Cha-
ex-
cep-
tions
raise
by
de-
stroy

1.29
-

Cha-
re-
turn
valu
of
ven-
dor_
and

drive
to
a
dic-
tio-
nary

1.30
-

Add
set_

and

get_

1.31

-

Add

Ver-

sion

Ob-

jects

in-

di-

rec-

tion

API

meth

ods:

obje

ob-

ject_

and

obje

1.32

-

Add

do_1

1.33

-

Add

up-

date

and

de-

stroy

port

grou

1.34

-

Add

hear

beat

1.35

-

Add
de-
stroy
and
up-
date

1.36

-

Add
cre-
ate_

1.37

-

Add
de-
stroy
and
up-
date

1.38

-

Add
vif_
vif_
vif_

1.39

-

Add
time
out
op-
tion
pa-
ram-
e-
ter
to

char

1.40

-

Add
in-
ject_

1.41

-

Add
cre-
ate_

1.42

-

Add
op-
tiona
ager
to
hear
beat

1.43

-

Add
do_r
do_r
and
can_

1.44

-

Add
add_
and
re-
mov

1.45

-

Add
con-

tinu

1.46

-

Add

re-

set_

to

up-

date

1.47

-

Add

sup-

port

for

con-

duc-

tor

grou

1.48

-

Add

al-

lo-

ca-

tion

API

1.49

-

Add

get_

and

ager

ar-

gu-

men

to

hear

beat

1.50

-

Add
set_
get_
and

get_

1.51
-

Add
ager
to
hear
beat

1.52
-

Add
de-
ploy
step
ar-
gu-
men
to
pro-
vi-
sion
ing

1.53
-

Add
dis-
able
to
do_1

1.54
-

Add
op-
tiona
ager
and

agen
to
hear
beat

RPC_API

add_node

Add
or
re-
plac
trait
for
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **tra**
a
list
of
trait
to
add
to
the
node
- **rep**
True
to
re-

plac
all
of
the
node
trait

- **top**
RPC
topic
De-
fault
to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod

Not-
Four
if
the
node
does
not
ex-
ist.

can_send

Re-
turn
when
the
RP-
CAF
sup-
port
the
cre-
ate_
meth

can_send

Re-
turn
when
the
RP-
CAF
sup-
port
node
res-
cue
meth
ods.

change_r

Cha
a
node
pow
state

Syn-
chro
ac-
quir
lock

node.

and
start
the
con-
duc-
tor
back
grou
task
to
char
pow
state
of
a

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **new**
one
of
iron
pow
state
val-
ues
- **tim**
time
out
(in
sec-
onds
pos-
i-
tive

indicates to use default timeout.

in-
te-
ger
(>
0)
for
any
pow
state
Non

- **top**
RPC
topic
De-
fault
to
self.

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn
task

continues

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
start
the
next
clear
ing

ductor for this RPC.

ac-
tion.
NOT
this
is
an
RPC
cast,
there
will
be
no
re-
spor
or
ex-
cep-
tion
raise
by
the
con-

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

continue

Sig-

ductor for this RPC.

nal
to
con-
duc-
tor
ser-
vice
to
start
the
next
de-
ploy
men
ac-
tion.
NOT
this
is
an
RPC
cast,
there
will
be
no
re-
spor
or
ex-
cep-
tion
raise
by
the
con-

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id

or
uuid

- **top**
RPC
topic
De-
fault
to
self.

create_

Cre-
ate
an
al-
lo-
ca-
tion.

Parame

- **con**
re-
ques
con-
text.

- **all**
an
al-
lo-
ca-
tion
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

create_

Syn-
chro

have
a
con-
duc-
tor
val
i-
date
and
cre-
ate
a
node
Cre-
ate
the
node
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
node
ob-
ject.

Parame

- **con**
re-
ques
con-
text.
- **nod**
a
cre-
ated
(but
not
save
node
ob-

ject.

- **top**
RPC
topic
De-
fault
to
self.

Returns

cre-
ated
node
ob-
ject.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
val-
i-
da-
tion
fails
for
any
dy-
nam
in-
ter-

faces (e.g. network_interface).

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
can

terfaces, and explicit values must be provided.

ductor will lock related node and trigger specific driver actions if they are needed.

be
cal-
cu-
latec
for
som
in-

create_p

Syn-
chro
have
a
con-
duc-
tor
val-
i-
date
and
cre-
ate
a
port
Cre-
ate
the
port
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
ob-
ject.
The
con-

Parame

-

con

re-ques-
con-text.

•

por

a cre-ated
(but not save
port ob-ject.

•

top

RPC topic
De-fault
to self.

Returns

cre-ated
port ob-ject.

destroy_

Dele-an-al-lo-ca-tion.

Parame

•

con
re-ques-
con-text.

•

all

perform deallocation.

an
al-
lo-
ca-
tion
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

Raises

In-
valid
State
if
the
as-
so-
ci-
ated
node
is
in
the
wron
pro-
vi-
sion
state
to

destroy.

Dele
a
node

Parame

- **con**
re-
ques
con-
text.

- **node**
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Nod
As-
so-
ci-
ated
if
the
node
con-
tains
an
in-
stan
as-
so-
ci-
ated
with
it.

Raises
In-

valid
State
if
the
node
is
in
the
wron
pro-
vi-
sion
state
to
per-
form
dele
tion.

destroy.

Dele
a
port

Parame

- **con**
re-
ques
con-
text.
- **por**
port
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
Loc
if

node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
port
does
not
ex-
ist.

destroy

Dele
a
port
grou

Paramete

- **con**
re-
ques
con-
text.
- **por**
port
grou
ob-
ject
-

top
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
port
grou
does
not
ex-
ist.

Raises

Port
grou
Not
if
port
grou
is
not
emp

destroy_

Dele
a
vol-
ume
con-
nec-
tor.

Dele
the
vol-
ume
con-
nec-
tor.

The
con-
duc-
tor
will
lock
the
re-
latec
node
dur-
ing
this

operation.

Parame

- **con**
re-
ques
con-
text

- **con**
vol-
ume
con-
nec-
tor
ob-
ject

- **top**

RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume

con-
nec-
tor
can-
not
be
foun

destroy_

Dele
a
vol-
ume
tar-
get.

Parame

- **con**
re-
ques
con-
text
- **tar**
vol-
ume
tar-
get
ob-
ject
- **top**
RPC
topi
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-

othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
tar-
get
does
not
ex-
ist

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

do_node_

Sig-
nal
to
con-
duc-
tor
ser-
vice
to

per-
form
man
ual
clear
ing
on
a
node

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
ID
or
UUID

- **cle**
a
list
of
clear
step
dic-
tio-
nar-
ies.

- **dis**
Whe
to
skip
boot
ing
ram
for
clear
ing.

- **top**
RPC

topic
De-
fault
to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
pow
driv
in-
ter-
face
faile

Raises

In-
valid
State
if
clea
ing
can
not
be
per-
form

Raises

Nod
Mai
te-
nan
if
node
is
in
main
te-
nan
mod

Raises

NodeLockError
if node is locked by another conductor.

Raises

NoFileNotFoundError
When there is no free world to start an asynchronous task.

do_node

Signal to conductor service to perform a deployment.

Parameters

- conductor

re-
ques
con-
text.

- **node**
node
id
or
uuid

- **rebu**
True
if
this
is
a
re-
buil
re-
ques

- **con.**
A
gzip
and
base
en-
code
con-
fig-
driv

- **top**
RPC
topic
De-
fault
to
self.

- **dep**
De-
ploy
step

Raises
In-
stan

ploy
Fail-
ure

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
fails

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

before this method is called.

task
The
node
must
already
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
un-
de-
ploy
state

do_node_

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
a
res-
cue.

Parame

- **con**
re-
ques-
con-
text.
- **nod**
node

vironment.

ID
or
UUID

- **res**
A
string
rep-
re-
sent
ing
the
pass
word
to
be
set
in-
side
the
res-
cue
en-

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
stan-
cue-
Fail-
ure

Raises
NoF
duc-
tor-
Wor
whe
ther
is
no
free
worl

method is called.

to
start
asyn
task

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
state
be-
fore
this

do_node

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
tear
dow
a
de-
ploy
men

Parame

- **con**
re-
ques
con-

text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

Raises
In-
stan-
ce
Fail-
ure

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu-
if
val-
i-
da-
tion
fails

Raises
Mis-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-

e-
ter
is
miss
ing

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn
task

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
de-
ploy
state
be-

fore this method is called.

do_node

Sig-
nal
to
con-
duc-
tor
ser-

vice
to
per-
form
an
un-
res-
cue.

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
ID
or
UUI

- **top**
RPC
topic
De-
fault
to
self.

Raises

In-
stan
Un-
res-
cue-
Fail-
ure

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free

method is called.

world
to
start
asyn
task
The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
state
be-
fore
this

do_provi

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
the
give
ac-
tion
on
a
node

Parame

-

con
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **act:**
an
ac-
tion.
One
of
iron

- **top:**
RPC
topic
De-
fault
to
self.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

Raises
NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task

Raises

In-
valid
State
if
the
re-
ques
ac-
tion
can
not
be
per-
form

This
en-
cap-
su-
lates
som
pro-
vi-
sion
ing
ac-
tions
in
a
sin-
gle
call.

driver_v

Pass
vend
spec
calls
whic
dont
spec
ify
a
node
to
a
driv

Han

cuted on a random conductor with the specified driver. If the method mode is async the conductor will start background worker to perform vendor action.

Parameters

- **conductor**
re-ques-
con-
text.
- **driver**
nam
of
the
driv
on
whic
to
call
the
meth
- **driver**
nam
of
the
ven-
dor
meth

for
use
by
the
drive

- **http**
the
HTT
meth
used
for
the
re-
ques

- **inf**
data
to
pass
thro
to
the
drive

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
for
pa-
ram-
e-
ter
er-
rors.

Raises
Miss

ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
driv
does
have
a
ven-
dor
in-
ter-
face
or

if the vendor interface does not support the specified `driver_method`.

Raises

Driv
Not-
Four
if
the
sup-
plie
driv
is
not
load

is invalid.

Raises

NoF
duc-
tor-
Wor
whe
ther
is
no
free
worl
to
start
asyn
task

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault
in-
ter-
face
for
a
hard
ware
type

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-

can be found for this drivers vendor interface.

face
im-
ple-
men-
ta-
tion

Returns

A
dic-
tio-
nary
con-
tain-
ing:

return

The
re-
spon-
se
of
the
in-
voke
ven-
dor
meth

async

Boo-
valu-
When
the
meth
was
in-
voke
asyn-
chro-
(Tru-
or
syn-
chro-
(Fal-
When
in-
voke
asyn

chronously the response will be always None.

attach

response object (True) or return it in the response body (False).

Boo
valu
Whe
to
at-
tach
the
re-
spor
of
the
in-
voke
ven-
dor
meth
to
the
HTT

get_boot

Get
the
cur-
rent
boot
de-
vice

Re-
turn
the
cur-
rent
boot
de-
vice
of
a
node

Paramete

- **con**
re-
ques
con-
text.
-

node
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men

Raises
In-
valid
Pa-
ram-

e-
ter-
Valu
whe
the
wron
driv
info
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plied
info.

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
iron
com
boo
or
Non
if
it

unknown.

is
un-
know

persist

When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_con

Get
the
con-
duc-
tor
whic
the
node
is
map
to.

Param

node

a
node
ob-
ject.

Returns

the
con-
duc-
tor
host

nam

Raises

No-
Valid
Host

get_cons

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises

Un-
sup-
port
ed-
Driv
ten-

sion
if
the
node
drive
does
sup-
port
con-
sole

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

get_curr

Get
RPC

topic
nam
for
the
cur-
rent
con-
duc-
tor.

get_driv

Get
the
prop
er-
ties
of
the
driv

Parame

- **con**
re-
ques
con-
text.
- **driv**
nam
of
the
driv
- **top**
RPC
topic
De-
fault
to
self.

Returns

a
dic-
tio-
nary
with
<pro

erty
nam
de-
scrip
tion:
en-
tries

Raises

Driv
Not-
Four

get_drive

Re-
triev
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the
give
drive

Parameters

- **con**
an
ad-
min
con-
text.
- **drive**
nam
of
the
drive
- **top**
RPC
topic
De-

fault
to
self.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not
have
ven-
dor
in-
ter-
face

Raises

Driv
Not-
Four
if
the
sup-
plie
driv
is
not
load

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault
in-
ter-
face

is invalid.

can be found for this drivers vendor interface.

for
a
hard
ware
type

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-
face
im-
ple-
men
ta-
tion

Returns

dic-
tio-
nary
of
<me
nam
meta
data
en-
tries

get_ind:

Get
node
hard
ware
com
po-
nent
in-
di-
ca-

for
state

Parameter

- **con**
re-
ques
con-
text.

- **node**
node
id
or
uuid

- **comp**
The
hard
ware
com
po-
nent
one
of
iron
com
comp

- **ind**
In-
di-
ca-
tor
IDs,
as
re-
port
by
get_

- **top**
RPC
topic
De-
fault
to

self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
driv
info
is
spec
i-
fied.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info.

Returns

In-
di-
ca-
tor
state
one
of
mod

get_node

Re-
triev
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the
give
node

Parame

- **con**
an
ad-
min
con-
text.

- **node**
the
id
or
uuid
of
a
node

- **top**
RPC
topic
De-
fault
to
self.

Returns

dic-
tio-
nary
of
<me
nam
meta
data
en-
tries

get_node

Re-
ques
the
node
from
the
con-
duc-
tor
with
an
agen
to-
ken

Parame

- **con**
re-

ques
con-
text.

- **node**
node
ID
or
UUID

- **topic**
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
Nod
ob-
ject
with
ager
to-
ken.

get_rai

Get
the
log-
i-
cal
disk
prop
er-

in the input RAID configuration.

ties
for
RAID
con-
fig-
u-
ra-
tion.
Gets
the
in-
for-
ma-
tion
about
log-
i-
cal
disk
prop-
er-
ties
which
can
be
spec-
i-
fied

Parame

- **con**
re-
ques-
con-
text.
- **dri**
nam
of
the
drive
- **top**
RPC
topic
De-
fault

to
self.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
driv
does
sup-
port
RAI
con-
fig-
u-
ra-
tion.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault
in-
ter-
face
for
a
hard
ware
type

is invalid.

Raises

No-
Vali
De-
fault
ForI
ter-

can be found for this drivers RAID interface.

cal disks and a textual description for them.

face
if
no
de-
fault
in-
ter-
face
im-
ple-
men-
ta-
tion

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
prop-
er-
ties
that
can
be
men-
tioned
for
log-
i-

get_ran

Get
an
RPC
topic
for
a
ran-
dom
con-
duc-
tor
ser-
vice

get_supp

Get the list of supported device

Return the list of supported boot device of a node

Parameters

- **context**
request context.
- **node_id**
node id or uuid
- **topic**
RPC topic. Default to self.

Raises

NodeLock

if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
whe
the
wron
driv
info
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-

e-
ter-
Valu
if
miss
ing
sup-
plic
info.

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fine
in
ironic
comm
boo

get_supp

Get
node
hard
ware
com
po-
nent
and
their
in-
di-
ca-
tors.

Parame

- **con**
re-
ques
con-
text.
-

node
node
id
or
uuid

- **comp**
The
hard
ware
com
po-
nent
one
of
iron
com
com

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the

node
drive
does
sup-
port
man
age-
men

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is
spec
i-
fied.

Raises

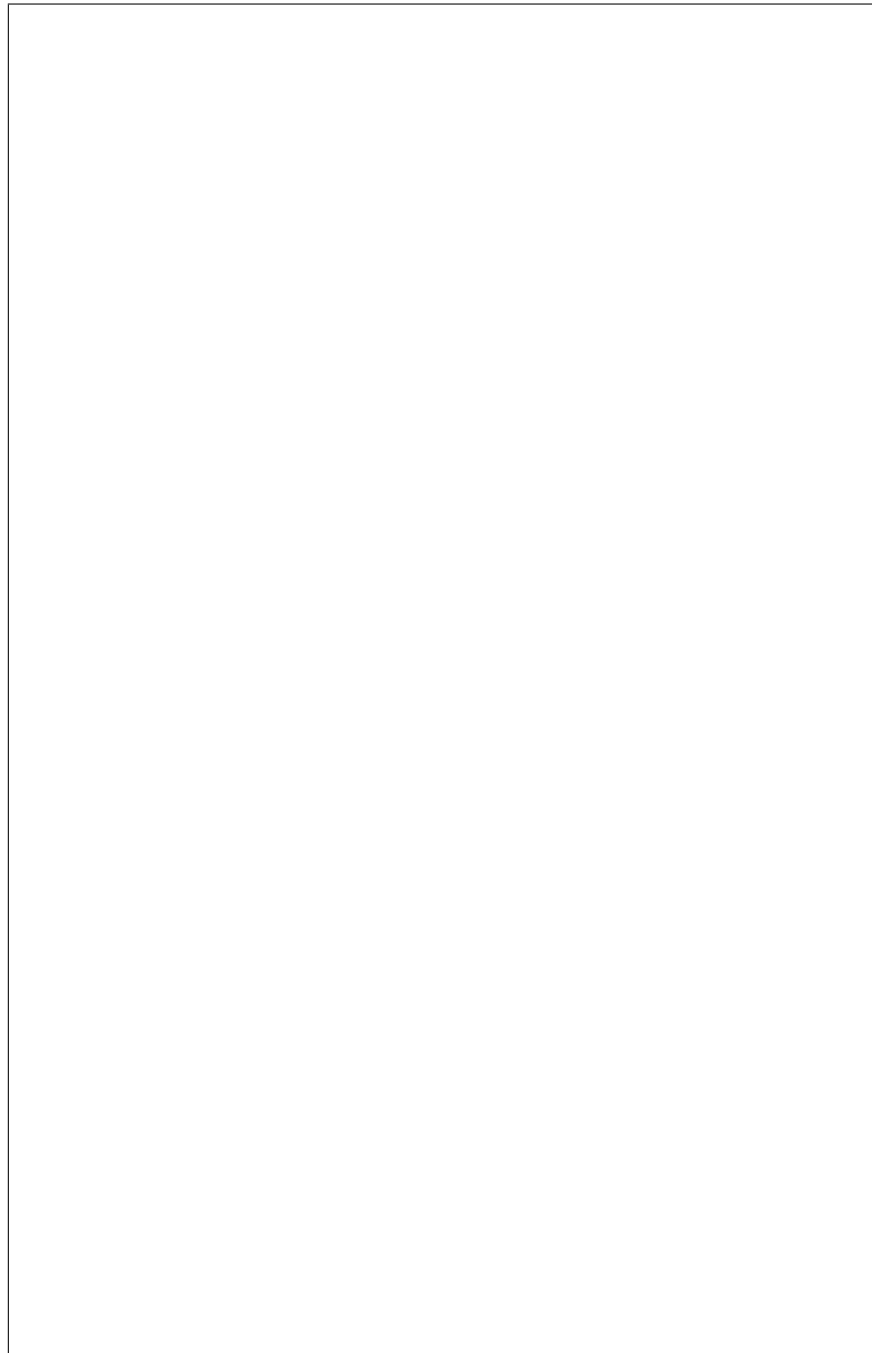
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plic
info.

Returns

A
dic-
tio-
nary
of
hard
ware
com

po-
nent
(*irc*
com
com
as
keys
with
in-
di-
ca-
tor

IDs as values.



get_top:

Get the RPC topic for the conductor service the node is mapped to.

Parameter
node

a node object.

Returns

an RPC topic string

Raises

No-ValidHosts

get_top:

Get the RPC topic name for a conductor support using the given driver

The
topic
is
used
to
route
mes-
sage
to
the
con-
duc-
tor
sup-
port
ing
the
spec
i-
fied

driver. A conductor is selected at random from the set of qualified conductors.

Paramete

driv
the
nam
of
the
driv
to
rout
to.

Returns

an
RPC
topic
strin

Raises

Driv
Not-
Four

heartbea

Pro-
cess
a
node
hear
beat

Parameter

- **con**
re-ques
con-text.
- **node**
node
ID
or
UUID
- **call**
URI
to
reach
back
to
the
range
- **topic**
RPC
topic
De-
fault
to
self.
- **age**
ran-
dom
gen-
er-
ated
val-
i-
da-
tion
to-
ken.
- **age**
the
ver-

sion
of
the
ager
that
is
hear
beat
ing

- **ager**
TLS
cer-
tifi-
cate
for
the
ager

- **ager**
The
sta-
tus
of
the
ager
that
is
hear
beat
ing

- **ager**
Op-
tiona
mes
sage
de-
scrib
ing
the
ager
sta-
tus

Raises
In-
valid
Pa-
ram-

e-
ter-
Valu
if
an
in-
valid
ager
to-
ken
is
re-
ceiv

inject_r

In-
ject
NM
for
a
node

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately
Be
awa
that
not

all drivers support this.

Parame

- **con**
re-
ques
con-

text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men
or
man

agement.inject_nmi.

valid boot device is specified.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
driv
info
is
spec
i-
fied
or
an
in-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info.

inspect_
Sig-
nals
the
con-
duc-
tor
ser-
vice
to
per-
form
hard
ware

in-
tro-
spec
tion.

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Har
ware
spec
tion-
Fail-
ure

Raises

NoF

duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn
task

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
in-
spec
tion.

Raises

In-
valid
State
if
in-
spec
is
not
a
valid
ac-
tion
to
do
in
the
cur-
rent

topic argument for this method.

state

object_a

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
in-
stan

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-
ten-
tiona
that
there
is
no

Paramete

- **con**
The
con-
text
with
whic
to
per-
form

the
ac-
tion

- **obj:**
The
ob-
ject
in-
stan-
ce
on
which
to
per-
form
the
ac-
tion

- **obj:**
The
name
of
the
ac-
tion
meth-
od
to
call

- **arg:**
The
po-
si-
tion
ar-
gu-
ment
to
the
ac-
tion
meth-

- **kw:**
The
key-
word
ar-

gu-
men
to
the
ac-
tion
meth

Raises

NotI
ple-
men
ed-
Er-
ror
whe
an
op-
er-
a-
tor
mak
an
er-
ror
dur-
ing
up-

grade

Returns

A
tu-
ple
with
the
up-
date
mad
to
the
ob-
ject
and
the
re-
sult
of
the
ac-

tion method

object_L

ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

Per-
form
a
back
port
of
an
ob-
ject
in-
stan-

The
de-
fault
be-
hav-
ior
of
the
base
Ver-
sion
dOb-
ject-
Se-
ri-
al-
izer,
upon
re-
ceiv-

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-
ten-
tiona

topic argument for this method.

that
there
is
no

Parameters

- **context**
The context with which to perform the backport.
- **obj**
An instance of a VersionedObject to be backported.
- **obj**
A dictionary of {obj: j-version mapping

Raises
Notl

grade

ple-
men
ed-
Er-
ror
r
whe
an
op-
er-
a-
tor
mak
an
er-
ror
dur-
ing
up-

Returns

The
dow
grad
in-
stan
of
ob-
jinst

object_c

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

We
wan
any
con-
duc-
tor
to
han-

topic argument for this method.

dle
this,
so
it
is
in-
ten-
tion:
that
there
is
no

Parameters

- **con**
The
con-
text
with
which
to
per-
form
the
ac-
tion
- **obj**
The
reg-
istry
name
of
the
ob-
ject
- **obj**
The
name
of
the
ac-
tion
meth
to
call

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kw**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Raises
Notl
ple-
men
ed-
Er-
ror
whe
an
op-
er-

grade

implementing VersionedObject class.

a-
tor
mak
an
er-
ror
dur-
ing
up-

Returns

The
re-
sult
of
the
ac-
tion
meth
whic
may
(or
may
not)
be
an
in-
stan
of
the

remove_r

Re-
mov
som
or
all
trait
from
a
node

Parame

- **con**
re-
ques
con-

removed from the node.

text.

- **node**
node
ID
or
UUID

- **traits**
a
list
of
traits
to
re-
mov
from
the
node
or
Non
If
Non
all
traits
will
be

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
does
not
ex-
ist.

Raises

Nod
Trai
Not-
Four
if
one
of
the
trait
is
not
foun

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node
Be
awa

support this.

that
not
all
drive

Parame

- **con**
re-
ques-
con-
text.

- **nod**
node
id
or
uuid

- **dev:**
the
boot
de-
vice
one
of
iro
com
boo

- **per:**
Whe
to
set
next
boot
or
mak
the
char
per-
ma-
nent
De-
fault
Fals

-

top
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
driv

valid boot device is specified.

info
is
spec
i-
fied
or
an
in-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info

set_cons

En-
able
the
con-
sole

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic

De-
fault
to
self.

- **enabl**
Boo
valu
whe
the
con-
sole
is
en-
able
or
dis-
able

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
con-
sole

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
driv
info
is
spec

i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn
task

set_ind:

Set
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
to

the
de-
sired
state

Parameter

- **con**
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **comp**
The
hard
ware
com
po-
nent
one
of
iron
comm
comp

- **ind**
In-
di-
ca-
tor
IDs,
as
re-
port
by
get_

- **sta**
In-
di-
ca-

tor
state
one
of
mod

- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
man
age-
men

Raises

In-
valid
Pa-
ram-

valid boot device is specified.

e-
ter-
Valu
whe
the
wron
driv
info
is
spec
i-
fied
or
an
in-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info.

set_targ

Stor
the
tar-
get
RAI
con-
fig-
u-
ra-
tion
on
the
node

Stor
the
tar-
get
RAI

con-
fig-
u-
ra-
tion
on
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
tar-
get
RAI
con-
fig-
u-
ra-
tion.
It
may
be
- **top**
RPC
topic
De-
fault
to

an empty dictionary as well.

tion.

self.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
RAI
con-
fig-
u-
ra-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
tar-
get
raid
con-
fig
fails

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

update_r

Syn-
chro
have
a
con-
duc-
tor
up-
date
the
node
in-
for-
ma-
tion.

Up-
date
the
node
in-
for-
ma-
tion
in

ductor will lock the node while it validates the supplied information. If `driver_info` is passed, it will be validated by the core drivers. If `instance_uuid` is passed, it will be set or unset only if the node is properly configured.

the
data
and
re-
turn
a
node
ob-
ject.
The
con-

Note
that
pow
shou
not
be
pass
via
this
meth
Use
char
for
ini-
ti-
at-
ing
driv
ac-
tion

Parameters

- **con**
re-
ques
con-
text.
- **nod**
a
char
(but
not
save

node
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

- **res**
whe
to
re-
set
hard
ware
in-
ter-
face
to
their
de-
fault

Returns
up-
date
node
ob-
ject,
in-
clud
ing
all
field

Raises
No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault

terfaces, and explicit values must be provided.

ductor will lock related node and trigger specific driver actions if they are needed.

can
be
cal-
cu-
late
for
som
in-

update_p

Syn-
chro
have
a
con-
duc-
tor
up-
date
the
port
in-
for-
ma-
tion.

Up-
date
the
port
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
ob-
ject.
The
con-

Parame

- **con**
re-
ques-
con-
text.

- **por**
a
char
(but
not
save
port
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

Returns

up-
date
port
ob-
ject,
in-
clud
ing
all
field

update_p

Syn-
chro
have
a
con-
duc-
tor
up-
date
the
port
grou

in-
for-
ma-
tion.

Up-
date
the
port
grou-
grou-
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
grou-
grou-
ob-
ject.

The conductor will lock related node and trigger specific driver actions if they are needed.

Parame

- **con**
re-
ques-
con-
text.
- **por**
a
chan
(but
not
save
port
grou-
ob-
ject.
- **top**
RPC
topi
De-

fault
to
self.

Returns

up-
date
port
grou
ob-
ject,
in-
clud
ing
all
field

update_v

Up-
date
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion.

Up-
date
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
vol-

ume connector object. The conductor will lock the related node during this operation.

Paramete

- **con**
re-
ques
con-
text
- **con**
a
char
(but
not
save
vol-
ume
con-
nec-
tor
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
con-
nec-
tors
UUI
is
be-
ing

chan

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor
can-
not

exists with the same values for type and connector_id fields

be
foun
Raises
Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
an-
othe
con-
nec-
tor
al-
read

Returns
up-
date
vol-
ume
con-
nec-
tor
ob-
ject,
in-
clud
ing
all
field

update_v

Up-
date
the
vol-
ume
tar-
gets
in-
for-
ma-

target object. The conductor will lock the related node during this operation.

tion.
Up-
date
the
vol-
ume
tar-
gets
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
vol-
ume

Parame

- **con**
re-
ques
con-
text
- **tar**
a
char
(but
not
save
vol-
ume
tar-
get
ob-
ject
- **top**
RPC
topic
De-
fault

to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
tar-
gets
UUI
is
be-
ing
char

Raises

Nod
Lock
if
the
node
is
al-
read
lock

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
does
not

ist

the same node ID and boot index values

ex-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

Returns

up-
date
vol-
ume
tar-
get
ob-
ject,
in-
clud

ing
all
field

validate

Val-
i-
date
the
core
and
stan
dara
ized
in-
ter-
face
for
drive

Paramete

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Returns

a
dic-
tio-
nary
con-
tain-

ing
the
re-
sults
of
each
in-
ter-
face
val-
i-
da-
tion.

vendor_p

Re-
ceiv
re-
ques
for
vend
spec
ac-
tions

Syn-
chro
val-
i-
date
driv
spe-
cific
info
or
get
driv
sta-
tus,
and
if
suc-
cess
ful
in-

vokes the vendor method. If the method mode is async the conductor will start background worker to perform vendor action.

Parame

-

con
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **dri**
nam
of
meth
for
driv

- **htt**
the
HTT
meth
used
for
the
re-
ques

- **inf**
info
for
node
driv

- **top**
RPC
topi
De-
fault
to
self.

Raises
In-
valid
Pa-
ram-
e-

ter-
Valu
if
sup-
plic
info
is
not
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not
have
ven-
dor
in-
ter-
face

Raises

NoF

duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn
task

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
dic-
tio-
nary
con-
tain-
ing:

return

The
re-
spor
of
the
in-
voke
ven-
dor
meth

async

Boo
valu

chronously the response will be always None.

response object (True) or return it in the response body (False).

When
the
meth
was
in-
voke
asyn
chro
(Tru
or
syn-
chro
(Fal
When
in-
voke
asyn

attach

Boo
valu
When
to
at-
tach
the
re-
spor
of
the
in-
voke
ven-
dor
meth
to
the
HTT

vif_atta

At-
tach
VIF
to
a
node

Parame

value is a unique identifier for that VIF.

- **con**
re-
ques-
con-
text.

- **nod**
node
ID
or
UUID

- **vif**
a
dic-
tio-
nary
rep-
re-
sent
ing
VIF
ob-
ject.
It
mus
have
an
id
key,
who

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
has
an
ex-

clu-
sive
lock
held
on
it

Raises

Net-
work
Er-
ror,
if
an
er-
ror
oc-
curs
dur-
ing
at-
tach-
ing
the
VIF.

Raises

In-
valid
Pa-
ram-
e-
ter
Valu-
if
a
pa-
ram-
e-
ter
that
re-
quir-
for
VIF
at-

tach is wrong/missing.

vif_det

De-
tach
VIF

from
a
node

Parameters

- **context**
re-ques
con-
text.

- **node**
node
ID
or
UUID

- **visibility**
an
ID
of
a
VIF.

- **topic**
RPC
topic
De-
fault
to
self.

Raises

NodeLockError
if
node
has
an
ex-
clu-
sive
lock
held
on
it

Raises

NetworkError

tach is wrong/missing.

world
Er-
ror,
if
an
er-
ror
oc-
curs
dur-
ing
de-
tach
ing
the
VIF.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
pa-
ram-
e-
ter
that
re-
quir
for
VIF
de-

vif_list

List
at-
tach
VIF
for
a
node

Parame

-

con
re-
ques
con-
text.

- **node**
node
ID
or
UUID

- **topic**
RPC
topic
De-
fault
to
self.

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

ID of the VIF.

Raises

Net-
work
Er-
ror,
if
an
er-
ror

oc-
curs
dur-
ing
list-
ing
the
VIF

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
pa-
ram-
e-
ter
that
re-
quir
for
VIF
list

is wrong/missing.

ironic.conductor.steps module

ironic.c

Find
an
iden
ti-
cal
step
in
the
list
of
step

ironic.c

Com
pare
step

ig-
nor-
ing
their
pri-
or-
ity.

ironic.c

Set
up
the
node
with
clear
step
in-
for-
ma-
tion
for
clear
ing.

For
au-
to-
mate
clear
ing,
get
the
clear
step
from
the
driv
For
man
ual
clear
ing,
the
user

clean steps are known but need to be validated against the drivers clean steps.

Paramet
dis
If
True
only
step

with
re-
quir
are
ac-
cept

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
there
is
a
prob
lem
with
the
user
clear
step

Raises

Nod
Clea
ing-
Fail-
ure
if
there
was
a
prob
lem
get-
ting
the
clear
step

ironic.c

Set
up
the
node
with
de-

ploy
men
step
in-
for-
ma-
tion
for
de-
ploy
ing.

Get
the
de-
ploy
step
from
the
drive

Paramet

res
Whe
to
re-
set
the
cur-
rent
step
to
the
first
one.

Raises

In-
stan-
ploy
Fail-
ure
if
there
was
a
prob-
lem
get-
ting
the
de-
ploy

men
step
ironic.c

Val-
i-
date
the
user
de-
ploy
step
and
the
de-
ploy
tem-
plate
for
a
node

Paramet

- **task**
A
Task
ager
ob-
ject
- **dep**
De-
ploy
step
to
val-
i-
date
Op-
tiona
If
not
pro-
vide
then
will
chec

internal info.

tion.

that are unsupported by the nodes driver interfaces or user deploy steps are unsupported by the nodes driver interfaces

node
drive

- **skip**
when
skip
missing
ing
step
that
are
not
yet
avail
able
at
the
time
of
val-
i-
da-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
in-
stan
has
trait
that
map
to
de-
ploy
step

Raises
In-
stan
ploy

driver.

`ironic.conductor.task_manager` module

Fail-
ure
if
there
was
a
prob
lem
get-
ting
the
de-
ploy
step
from
the

A
con-
text
man
ager
to
per-
form
a
se-
ries
of
task
on
a
set
of
re-
sour

Tas.
is
a
con-
text
man
ager
cre-
ated
on-
dem

a node and its resources.

tion that the TaskManager instance exists. You may create a TaskManager instance without locking by passing `shared=True` when creating it, but certain operations on the resources held by such an instance of TaskManager will not be possible. Requiring this exclusive lock guards against parallel operations interfering with each other.

to
al-
low
syn-
chro
nize
ac-
cess
to

The
Tas.
will.
by
de-
fault
ac-
quir
an
ex-
clu-
sive
lock
on
a
node
for
the
du-
ra-

A
shar
lock
is
use-
ful
whe
per-
form
ing
non-
inter
op-
er-
a-
tions

dating the driver interfaces.

instances, that are typically deployed on different hosts.

invocation requires an exclusive lock.

such
as
val-
i-

An
ex-
clu-
sive
lock
is
store
in
the
data
to
co-
or-
di-
nate
be-
twee
iron
con
man

Tas.
meth
ods,
as
well
as
driv
meth
ods,
may
be
dec-
o-
rate
to
de-
ter-
min
whe
their

The
Task
ager

in-
stan-
ex-
pose
cer-
tain
node
re-
sour-
and
prop-
er-
ties
as
at-
tribu-
that

you may access:

task.cont
The
con-
text
pass
to
Task
ager

task.share
Fals
if
Nod
is
lock
True
if
it
is
not
lock
(The
share
kwa
arg
of
Task
ager

task.node
The
Nod
ob-
ject

task.port
Port
be-
long
ing
to
the
Nod

task.port
Port
grou
be-
long
ing
to
the
Nod

task.volu
Stor
age
con-
nec-
tors
be-
long
ing
to
the
Nod

task.volu
Stor
age
tar-
gets
as-
sign
to
the
Nod

task.driv
The
Driv
for
the
Nod
or
the
Driv
base

on
the
driv
kwa
of
Task
ager

Ex-
am-
ple
us-
age:

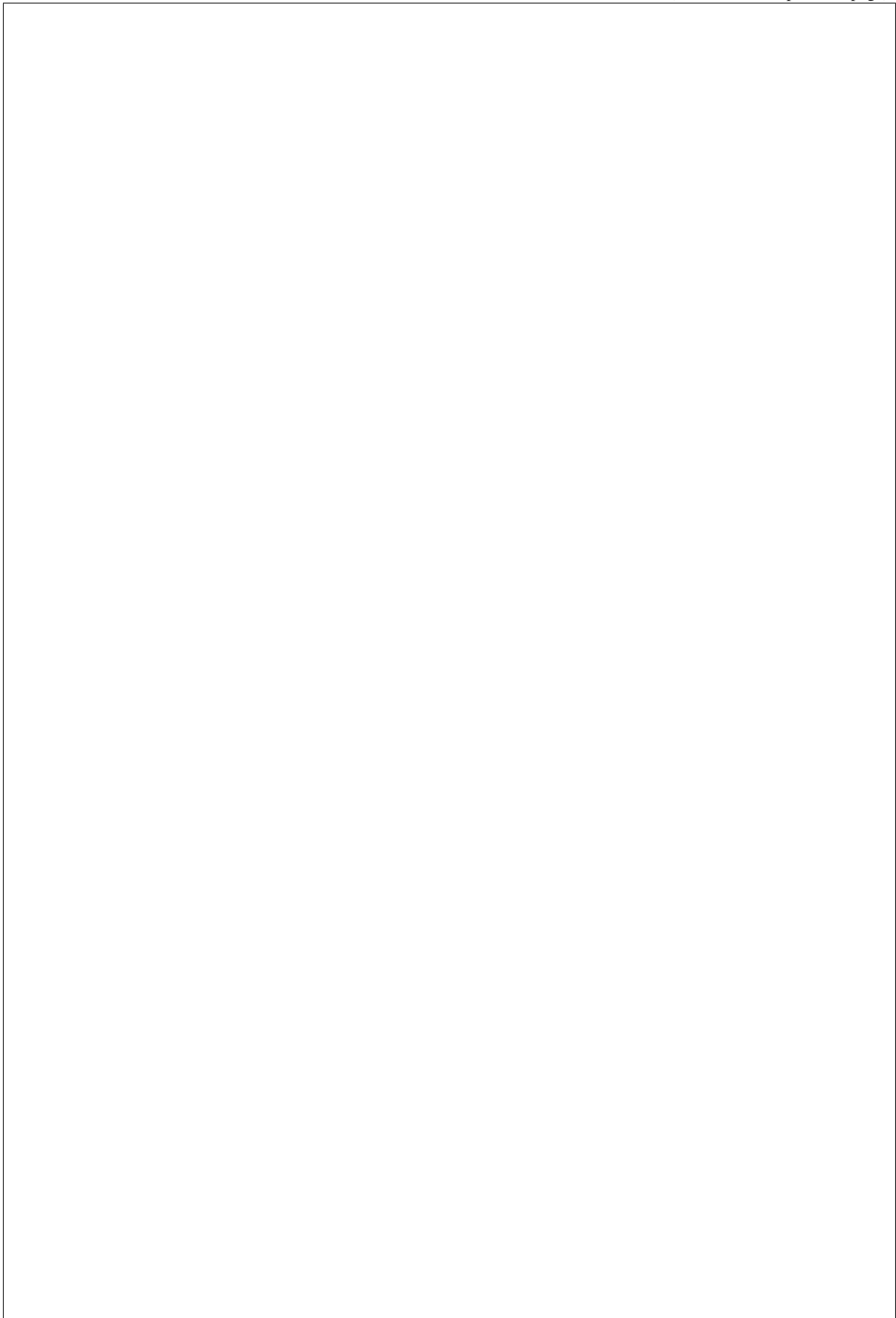


witl
→ t
→ m
→ a
→
→ n
→ i
→
→ p
→ 'j
→ O
→ 'a
→ a
→ t

→
→
→
→ t
→ d
→ p
→ p
→ o
→ n

If
you
need
to
ex-
e-
cute
task
requ
code
in
a
back

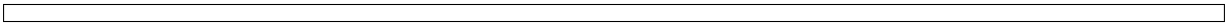
(continued from previous page)



(continues on next page)

witl
→ t
→ m
→ a
→ u
→ n
→ i
→ u
→ p
→ 'l
→ W
→ 'l
→ a
→ t
u
→ u
→ u
→ u
→
→ <
→ S
→ W
→
u
→ u
→ u
→ u
→ t
→ S
→ s
→ e
→ h
→ e
u
→ u
→ u
→ u
→ t
→ S
→ a
→ _
→ S
→ W
→
u
→ u
→ u
→ u
→ u
→ u
→ u
→ u

(continued from previous page)



Node and Ports) when beginning a unit of work.

class ir

Base
obj

Con
text
man
ager
for
task

This
class
wrap
the
lock
ing,
driv
load
ing,
and
ac-
qui-
si-
tion
of
re-
latec
re-
sour
(eg,

downgra

Dow
grad
the
lock
to
a
shar
one.

load_dr

property

property

property

process

Pro-
cess
the
give
even
for
the
task
cur-
rent
state

Parame

- **even**
the
nam
of
the
even
to
pro-
cess
- **cal.**
op-
tiona
call-
back
to
in-
voke
upon
even
tran-
si-
tion
- **cal.**
op-
tiona

are no workers available (`err_handler` should accept arguments `node`, `prev_prov_state`, and `prev_target_state`)

args
to
pass
to
the
call-
back
meth

- **cal**
op-
tion
kwa
to
pass
to
the
call-
back
meth

- **err**
op-
tion
er-
ror
han-
dler
to
in-
voke
if
the
call-
back
fails
eg.
be-
caus
there

- **tar**
if
spec
i-
fied,
the
tar-

the target state from the fsm

get
pro-
vi-
sion
state
for
the
node
Oth-
er-
wise
use

Raises

In-
valid
State
if
the
even
is
not
al-
lowe
by
the
as-
so-
ci-
ated
state
ma-
chin

release

Un-
lock
a
node
and
re-
lease
re-
sour
If
an
ex-
clu-
sive
lock

that this instance of TaskManager should no longer be accessed.

is
held
un-
lock
the
node
Re-
set
at-
tribu
to
mak
it
clear

set_spar

Cre-
ate
a
hool
to
han-
dle
ex-
cep-
tion
when
spav
ing
a
task

Cre-
ate
a
hool
that
gets
called
upon
an
ex-
cep-
tion
be-
ing
raise
from
spav
ing

ground thread to do a task.

ject that was raised.

a
back

Parame

- **_on.**
a
calla
ob-
ject,
its
first
pa-
ram-
e-
ter
shou
ac-
cept
the
Ex-
cep-
tion
ob-
- **arg**
ad-
di-
tiona
args
pass
to
the
calla
ob-
ject.
- **kwa.**
ad-
di-
tiona
kwa
pass
to
the
calla
ob-

ject.

spawn_a:

Call this to spawn a thread to complete the task.

The specified method will be called when the TaskManager instance executes its.

Parameters:

- **_spawn**
a method that returns a Greenthread object
- **args**
args passed to

the
meth

- **kwargs**
ad-
di-
tion:
kwargs
pass
to
the
meth

upgrade_

Up-
grad
a
shar
lock
to
an
ex-
clu-
sive
lock

Also
relo:
node
ob-
ject
from
the
data
If
lock
is
al-
read
ex-
clu-
sive
only
char
the
lock

purpose when provided with one.

Parame

-

purp
op-
tion-
ally
char
the
pur-
pose
of
the
lock

- **ret:**
whe
to
retry
lock
ing
if
it
fails
the
class
leve
valu
is
used
by
de-
fault

Raises
Nod
Lock
if
an
ex-
clu-
sive
lock
re-
main
on
the
node
af-
ter
node

property

property

`ironic.c`

Shor
cut
for
ac-
quir-
ing
a
lock
on
a
Nod

Paramet

con
Re-
ques
con-
text.

Returns

An
in-
stan
of
Tas

`ironic.c`

Dec
o-
ra-
tor
to
re-
quir
an
ex-
clu-
sive
lock

Dec
o-
rate
func
tions
mus
take
a

TaskManager
as
the
first
pa-
ram-
e-
ter.
Dec
o-
rate
class

methods should take a *TaskManager* as the first parameter after self.

ironic.conductor.utils module

ironic.c

Set
node
state
when
a
task
was
abon
due
to
con-
duc-
tor
take
over

Parameter

task
a
Task
ager
in-
stan

ironic.c

Add
a
se-
cret
to-
ken
to
driv

for
IPA
ver-
i-
fi-
ca-
tion.

Paramet

- **node**
Nod
ob-
ject
- **pre**
Boo
valu
de-
fault
Fals
whic
in-
di-
cate
if
the
to-
ken
shou
be
marl
as
pre-

generated in order to facilitate virtual media booting where the token is embedded into the configuration.

`ironic.c`

Che
that
the
agen
is
likel
alive

The
meth
then
chec
for

by `[deploy]fast_track_timeout`, then agent is presumed alive.

the
last
ager
hear
beat
and
if
it
oc-
cure
with
the
time
out
set

Paramet

- **node**
A
node
ob-
ject.
- **time**
Hea
beat
time
out,
de-
fault
to
fast_

ironic.c

Buil
a
con-
fig-
driv
from
pro-
vide
meta
net-
worl
and
user

accordingly.

If
uuid
or
nam
are
not
pro-
vide
in
the
meta
they
de-
faul
to
the
node
uuid
and
nam

Paramet

- **node**
an
Iron
node
ob-
ject.
- **con**
A
con-
fig-
driv
as
a
dict
with
keys
met.
net
use.
and
ven
(all
op-
tion

Returns

A
gzip
and
base
en-
code
con-
fig-
driv
as
a
strin

ironic.c

Put
a
faile
node
in
CLE
FAIL
and
main
te-
nan
(if
need

Paramet

- **task**
a
Task
ager
in-
stan
- **log**
Mes
sage
to
be
logg
- **err**

Message for the user. Option if not provide *login* is used

- **trace**
When to log a traceback back. Default to False

- **teardown**
When to clean up the PXE and DHCP files after cleaning. Default to True

- **set**
When to set

only if a clean step is being executed on a node.

node
to
failed
state
De-
fault
to
True

- **set.**
When
to
set
main
te-
nan-
mod
If
Non
main
te-
nan-
mod
will
be
set
if
and

`ironic.c`

Clea
de-
ploy
task
af-
ter
time
out.

Paramet

task
a
Task
ager
in-
stan-

`ironic.c`

Clea
a

clearing
task
after
time
out.

Parameter

task
a
Task
ager
in-
stan

ironic.c

Clear
res-
cue
task
af-
ter
time
out.

Parameter

task
a
Task
ager
in-
stan

ironic.c

Put
a
failed
node
in
DE-
PLC
FAIL

Parameter

- **task**
the
task
-

log
mes
sage
to
be
logg

- **err**
mes
sage
for
the
user

- **tra**
Boo
True
to
log
a
trace
back

- **cle**
Boo
True
to
clea
up

`ironic.c`

A
dec-
o-
ra-
tor
for
fail-
ing
op-
er-
a-
tion
on
fail-
ure.

`ironic.c`

Che

ations if we already have a ramdisk heartbeating through external means.

if
the
op-
er-
a-
tion
can
be
a
strea
line
de-
ploy
men
se-
quer

This
is
main
fo-
cuse
on
en-
sur-
ing
that
we
are
able
to
quic
se-
quer
thro
op-
er-

Paramet

task

Task

ager

ob-

ject

Returns

True

if

[de-

ploy

is

set

no `last_error` is present for the node indicating that there was a recent failure.

to
True
no
iSCSI
boot
con-
fig-
u-
ra-
tion
is
pres
and

`ironic.c`

Get
any
at-
tach
vif
ID
for
the
port

Parameter

port
The
port
ob-
ject
upon
which
to
check
for
a
vif
reco

Returns

Re-
turn
a
tu-
ple
of
the
vif
if
foun

of a string, tenant, cleaning provisioning, rescuing.

port.

and
the
use
of
the
vif
in
the
form

Raises

In-
valid
State
ex-
cep-
tion
upon
find-
ing
a
port
with
a
tran-
sien
state
vif
on
the

ironic.c

Get
con-
fig-
drive
as
an
ISO
im-
age
or
a
URI
Con-
verts
the
JSO
rep-

turned unchanged.

re-
sen-
ta-
tion
into
an
im-
age.
URI
and
raw
con-
tents
are
re-

Paramet

node
an
Iron
node
ob-
ject.

Returns

A
gzip
and
base
en-
code
con-
fig-
driv
as
a
strin

`ironic.c`

`ironic.c`

`ironic.c`

Has
a
sup-
plied
pass
wor

Paramet

pas
pass
wor
to
be
hash

ironic.c

De-
ter-
min
if
the
to-
ken
was
gen-
er-
ated
for
out
of
band
con-
fig-
u-
ra-
tion.

Iron
sup-
port
the
abil-
ity
to
pro-
vide
con-
fig-
u-
ra-
tion
data
to
the
ager
thro
the

a virtual floppy or as part of the virtual media image which is attached to the BMC.

This

token prior to rebooting the token. This is important as tokens provided through out of band means persist in the virtual media image, are loaded as part of the agent ramdisk, and do not require regeneration of the token upon the initial lookup, ultimately making the overall usage of virtual media and pregenerated tokens far more secure.

False in all other cases.

metl
help
us
iden
tify
WH
we
did
so
as
we
dont
need
to
re-
mov
reco
of
the

Paramet
nod
Nod
Ob-
ject

Returns
True
if
the
to-
ken
was
pre-
gen-
er-
ated
as
in-
di-
cate
by
the
node
driv
field

ironic.c

De-
ter-
min-
if
an
ager
to-
ken
is
pres
upon
a
node

Paramet
nod
Nod
ob-
ject

Returns
True
if
an
ager
valu
is
pres
in
a
node
driv
field

`ironic.c`

Val-
i-
date
if
a
sup-
plied
to-
ken
is
valid
for
the
node

Paramet

- **node**
Node
ob-
ject
- **token**
A
to-
ken
valu
to
val-
i-
date
agai
the
driv
field
ager

Returns

True
if
the
sup-
plie
to-
ken
mat
the
to-
ken
reco
in
the
sup-
plie
node
ob-
ject.

`ironic.c`

Che
a
fast
track
is
avai
able

perform a fast track sequence meaning that we already have a ramdisk running through another means like discovery. If not valid, False is returned.

by `[deploy]fast_track_timeout` and the power state for the machine is `POWER_ON`, then fast track is permitted.

This
meth
first
en-
sure
that
the
node
and
con-
duc-
tor
con-
fig-
u-
ra-
tion
is
valid
to

The
meth
then
chec
for
the
last
ager
hear
beat
and
if
it
oc-
cure
with
the
time
out
set

Paramet
task
Task
ager
ob-
ject

Returns

True
if
the
last
heard
beat
that
was
reco
was
with
the
[de-
ploy
set-
ting

`ironic.c`

Gen
er-
ate
a
ran-
dom
salt
with
the
in-
di-
ca-
tor
tag
for
pass
wor
type

Returns

a
valid
salt
for
use
with
cryp

`ironic.c`

Do
cach
of

bios
set-
tings
if
sup-
port
by
drive

ironic.c

Cach
the
ven-
dor
if
it
can
be
de-
tecte

ironic.c

Reac
cur-
rentl
set
boot
mod
from
a
node

Reac
the
boot
mod
for
a
node
If
boot
mod
cant
be
dis-
cov-
ered
Non
is
re-
turn

Parameter

task

a

Task

ager

in-

stan

Raises

Drive

Op-

er-

a-

tion

or

its

deri

tive

in

case

of

drive

run-

time

er-

ror.

Raises

Un-

sup-

port

ed-

Drive

ten-

sion

if

cur-

rent

drive

does

not

have

man

age-

men

in-

ter-

face or `get_boot_mode()` method is not supported.

Returns

Boo

mod

One

of
iron
com
boo
or
Non
if
boot
mod
cant
be
dis-
cov-
ered

ironic.c

Cha
pow
state
or
re-
set
for
a
node
Per-
form
the
re-
ques
pow
ac-
tion
if
the
tran-
si-
tion
is
re-
quir

Paramet

- **task**
a
Task
ager
in-

stand
con-
tain-
ing
the
node
to
act
on.

- **new**
Any
pow
state
from
iron

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

indicates to use default timeout.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
state
is

info is specified.

upon setting power on.

action.

spec
i-
fied
or
the
wron
driv

Raises

Stor
ageE
ror
whe
a
fail-
ure
oc-
curs
up-
dat-
ing
the
node
stor-
age
in-
ter-
face

Raises

othe
ex-
cep-
tions
by
the
node
pow
driv
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

ironic.c

Set
the
boot
de-
vice
for
a
node

If
the
node
that
the
boot
de-
vice
char
is
be-
ing
re-
ques
for
is
in
ADC
ING
state

the boot device will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

Paramet

- **task:**
a
Task
ager
in-
stan
- **dev:**
Boo
de-
vice
Val-
ues
are

vend
spec

- **per:**
Whe
to
set
next
boot
or
mak
the
char
per-
ma-
nent
De-
fault
Fals

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion
of
the
Man
age-
men
ter-

face fails.

ironic.c

Set
the
boot
mod
for
a
node

interface.

boot mode will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

Sets
the
boot
mod
for
a
node
if
the
node
driv
in-
ter-
face
con-
tains
a
man
age-
men

If
the
node
that
the
boot
mod
char
is
be-
ing
re-
ques
for
is
in
ADC
ING
state
the

Paramet

- **task**
a
Task
ager

in-
stan

- **mod**
Boo
mod
Val-
ues
are
one
of
iro
com
boo

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion
of
the
Man
age-
men
ter-

face fails.

Raises

Driv
Op-
er-
a-
tionl
or
its
deri
tive
in
case
of
driv
run-

or method is unsupported.

time
er-
ror.
Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not
have
ven-
dor
in-
ter-
face

ironic.c

Wait
for
node
to
be
in
new
pow
state

Paramet

- **task**
a
Task
ager
in-
stan
- **new**
the
de-
sired

the conductor.power_state_change_timeout config value.

new
pow
state
one
of
the
pow
state
in
iro
com
sta

- **time**
num
ber
of
sec-
onds
to
wait
be-
fore
giv-
ing
up.
If
not
spec
i-
fied,
uses

Raises

Pow
er-
State
Fail-
ure
if
time
out

ironic.c

ironic.c

ironic.c

No-
tify

the
con-
duc-
tor
to
re-
sum
an
op-
er-
a-
tion.

Paramet

- **task**
the
task
- **oper**
the
op-
er-
a-
tion,
a
strin

ironic.c

Pow
ers
on
node
if
it
is
pow
ered
off
and
has
a
Sma
NIC
port

Paramet

task
A
Task

ager
ob-
ject

Returns

the
pre-
vi-
ous
pow
state
or
Non
if
no
char
were
mad

Raises

ex-
cep-
tion.
if
ager
sta-
tus
didn
mat
the
re-
quir
sta-
tus
af-
ter
max
retry
at-

tempts.

ironic.c

Set
the
node
pow
state
if
er-
ror
oc-
curs

This
hool
gets
called
upon
an
ex-
cep-
tion
be-
ing
raise
when
spaw
ing
the
world
threa
to
char

the power state of a node.

Paramet

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **node**
an
Iron
node
ob-
ject.
- **power**
the
pow
state
to
set
on

reconfiguration, then restores the power state.

the
node
ironic.c

Hand
dle
the
pow
state
for
a
node
re-
con-
fig-
u-
ra-
tion.

Power
ers
the
node
on
if
and
only
if
it
has
a
Sma
NIC
port
Yiel
for
the
ac-
tual

Paramet
task
A
Task
ager
ob-
ject.

ironic.c

Set

provisioning to a node like deployment, tear down, or cleaning.

the
node
pro-
vi-
sion
ing
state
if
er-
ror
oc-
curs

This
hool
gets
calle
upon
an
ex-
cep-
tion
be-
ing
raise
whe
spav
ing
the
worl
to
do
som

Paramet

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **nod**
an

Iron
node
ob-
ject.

- **pro**
the
pro-
vi-
sion
state
to
be
set
on
the
node

- **tar**
the
tar-
get
pro-
vi-
sion
state
to
be
set
on
the
node

ironic.c

Help
to
re-
mov
the
agen
reco

ironic.c

Help
to
re-
mov
res-
cue
pass

then caller needs to explicitly indicate it.

wor
from
a
node

Re-
mov
res-
cue
pass
wor
from
node
It
save
node
by
de-
fault
If
node
shou
not
be
save

Paramet

- **node**
an
Iron
node
ob-
ject.
- **save**
Boo
True
(de-
fault
to
save
the
node
Fals
oth-
er-
wise

ironic.c

Clea
res-
cue
task
af-
ter
time
out
or
fail-
ure.

Paramet

- **task**
a
Task
ager
in-
stan
- **msg**
a
mes
sage
to
set
into
node
last_
field
- **set**
a
bool
flag
to
in-
di-
cate
if
node
need
to
be
tran-
si-
tion

state. By default node would be transitioned to a failed state.

to
a
failed

ironic.c

Cha
the
node
pow
state
if
pow
is
not
Non

Paramet

- **task**
A
Task
ager
ob-
ject
- **pow**
pow
state

ironic.c

Che
if
node
clea
ing
need
to
be
skip
for
an
spe-
cific
node

Paramet

nod
the
node

to
con-
side

ironic.c

Han
dle
spav
ing
er-
ror
for
node
clea
ing.

ironic.c

Han
dle
spav
ing
er-
ror
for
node
de-
ploy
ing.

ironic.c

Han
dle
spav
ing
er-
ror
for
node
res-
cue.

ironic.c

Stor
cer-
tifi-
cate
re-
ceiv
from
the

ager
and
re-
turn
its
path

ironic.c

Cal-
cu-
late
the
next
step
in-
dex
and
up-
date
the
node

Paramet

- **task**
A
Task
ager
ob-
ject

- **step**
The
type
of
step
to
pro-
cess
clear
or
de-
ploy

Returns

In-
dex
of
the
next

step
ironic.c

Val-
i-
date
trait
in
in-
stan

All
trait
in
in-
stan
mus
also
ex-
ist
as
node
trait

Parameter
node
an
Iron
node
ob-
ject.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
in-
stan
trait
are
badl
for-
mat-
ted,
or
con-

tain traits that are not set on the node.

ironic.c

Val-
i-
date
the
con-
sis-
tenc
of
phys
i-
cal
net-
worl
of
port
in
a
port
grou

Val-
i-
date
the
con-
sis-
tenc
of
a
port
phys
i-
cal
net-
worl
with
othe
port
in
the

same portgroup. All ports in a portgroup should have the same value (which may be None) for their `physical_network` field.

Dur-
ing
cre-
ation
or
up-

ing validation criteria:

raise PortgroupPhysnetInconsistent. This shouldnt ever happen.

date
of
a
port
in
a
port
grou
we
ap-
ply
the
fol-
low-

- If the port group has existing port with different physical network we

- If the port has a physical network that is in-

in the portgroup, we raise exception.Conflict.

ment mapping algorithm should operate in a legacy (physical network unaware) mode for this port or portgroup. This allows existing ironic nodes to continue to function after an upgrade to a release including physical network support.

con-
sis-
tent
with
othe
port

If
a
port
phys
i-
cal
net-
worl
is
Non
this
in-
di-
cate
that
iron
ics
VIF
at-
tach

Paramet

- **task**
a
Task
ager
in-
stan
- **por**
a
port
ob-
ject
to
be
val-
i-

ent physical network.

signed the same physical network.

date
Raises
Con
flict
if
the
port
is
a
men
ber
of
a
port
grou
whic
is
on
a
dif-
fer-

Raises
Port
grou
Phys
net-
Inco
sis-
tent
if
the
port
port
grou
has
port
whic
are
not
all
as-

ironic.c

Che
if
the
time
is

with
the
pre-
vi-
ous
time
out
sec-
onds
from
now

Parameter

- **value**
a
string
rep-
re-
sent-
ing
date
and
time
or
Non

- **time**
time
out
in
sec-
onds

`ironic.c`

Re-
mov-
tem-
po-
rary
clear
ing
field
from
drive

`ironic.c`

Re-
mov

tem-
po-
rary
de-
ploy
men
field
from
drive

`ironic.c`

Wip
in-
for-
ma-
tion
that
shou
not
sur-
vive
re-
boot
off.

`ironic.c`

Re-
mov
agen
URI
and
to-
ken
from
the
task

Module contents

`ironic.conf` package

Submodules

`ironic.conf.agent` module

`ironic.c`

ironic.conf.anaconda module

ironic.c

ironic.conf.ansible module

ironic.c

ironic.conf.api module

ironic.c

ironic.conf.audit module

ironic.c

ironic.conf.auth module

ironic.c

Add
auth
op-
tion
to
sam
ple
con-
fig
As
thes
are
dy-
nam
i-
cally
reg-
is-
tere
at
run-
time
this
adds

auth_plugins when generating sample config.

istered at runtime depending on auth plugin used.

ironic.conf.cinder module

op-
tions
for
mos
used

ironic.c

Reg-
is-
ter
sess

and
auth
relat
op-
tions

Reg-
is-
ters
only
ba-
sic
auth
op-
tions
shar
by
all
auth
plu-
g-
ins.
The
rest
are
reg-

ironic.c
ironic.c

ironic.conf.conductor module

ironic.c

ironic.conf.console module

ironic.c

ironic.conf.database module

ironic.c

ironic.conf.default module

ironic.c

ironic.c

ironic.conf.deploy module

ironic.c

ironic.conf.dhcp module

ironic.c

ironic.conf.drac module

ironic.c

ironic.conf.glance module

ironic.c

ironic.c

ironic.conf.healthcheck module

ironic.c

ironic.conf.ibmc module

ironic.c

ironic.conf.ilo module

ironic.c

ironic.conf.inspector module

ironic.c

ironic.c

ironic.conf.ipmi module

ironic.c

ironic.conf.irmc module

ironic.c

ironic.conf.metrics module

ironic.c

ironic.conf.metrics_statsd module

ironic.c

ironic.conf.molds module

ironic.c

ironic.conf.neutron module

ironic.c

ironic.c

ironic.conf.nova module

ironic.c

ironic.c

ironic.conf.opts module

ironic.c

Re-
turn
a
list
of
oslo
op-
tion
avai-
able
in
Iron
code

The
re-
turn
list
in-
clud
all
oslo
op-
tion
Each

ple. The first element is the name of the group, the second element is the options.

el-
e-
men
of
the
list
is
a
tu-

The
func
tion
is
dis-
cov-
er-
able
via
the
iron.
en-
try
point
un-
der
the
oslo
nam
pace

The
func
tion
is
used
by
Oslo
sam-
ple
con-
fig
file
gen-
er-
a-
tor
to
dis-
cove
the

options.

Returns

a
list
of
(gro
op-
tion
tu-
ples

ironic.c

ironic.conf.pxe module

ironic.c

ironic.conf.redfish module

ironic.c

ironic.conf.service_catalog module

ironic.c

ironic.c

ironic.conf.snmp module

ironic.c

ironic.conf.swift module

ironic.c

ironic.c

`ironic.conf.xclarity` module

`ironic.c`

Module contents

`ironic.db` package

Subpackages

`ironic.db.sqlalchemy` package

Submodules

`ironic.db.sqlalchemy.api` module

SQL
stor-
age
back
end.

class `ironic.db.sqlalchemy.api`
Base
ironic.db.sqlalchemy.api
Connection
SqlA
con-
nec-
tion.

add_node
Add
tag
to
the
node
If
the
node
and
tag
pair

al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parame

- **node**
The
id
of
a
node

- **tag**
A
tag
strin

Returns

the
Nod
Tag
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

add_node

Add
trait
to
the
node

If
the

node
and
trait
pair
al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parameter

- **node**
The
id
of
a
node
- **trait**
A
trait
string
- **version**
the
ver-
sion
of
the
ob-
ject.

Returns

the
Nod
Trai
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-

ter-
Valu
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
Not-
Four
if
the
node
is
not
foun

check_no

Che
a
list
of
node
iden
ti-
ties
and
map
it
to
UUI

This
call
take
a
list
of
node
nam
and/
UUI
and

early if any identities cannot possible be used as names or UUIDs.

tries
to
con-
vert
then
to
UUID
It
fails

Parameters

identities
List
of
iden-
ti-
ties.

Returns

A
map-
ping
from
re-
ques-
iden-
ti-
ties
to
node
UUID

Raises

Node-
Not-
Found
if
some
iden-
ti-
ties
were
not
found
or
can-
not
be
valid
nam-
or
UUID

check_ve

Che
the
who
data
for
in-
com
pat-
i-
ble
ob-
jects

This
scan
all
the
ta-
bles
in
sear
of
ob-
jects
that
are
not
sup-
port
i.e.,
thos
that
are

not specified in *ironic.common.release_mappings.RELEASE_MAPPING*. This includes objects that have null version values.

Parame

igno
List
of
mod
nam
to
skip

Returns

A
Boo
True
if
all

the
ob-
jects
have
sup-
port
ver-
sion
Fals
oth-
er-
wise

clear_no

clear_no

create_a

Cre-
ate
a
new
al-
lo-
ca-
tion.

Parame

valu

Dict
of
val-
ues
to
cre-
ate
an
al-
lo-
ca-
tion
with

Returns

An
al-
lo-
ca-
tion

Raises

Al-

lo-
ca-
tion
pli-
cate
Nam

Raises

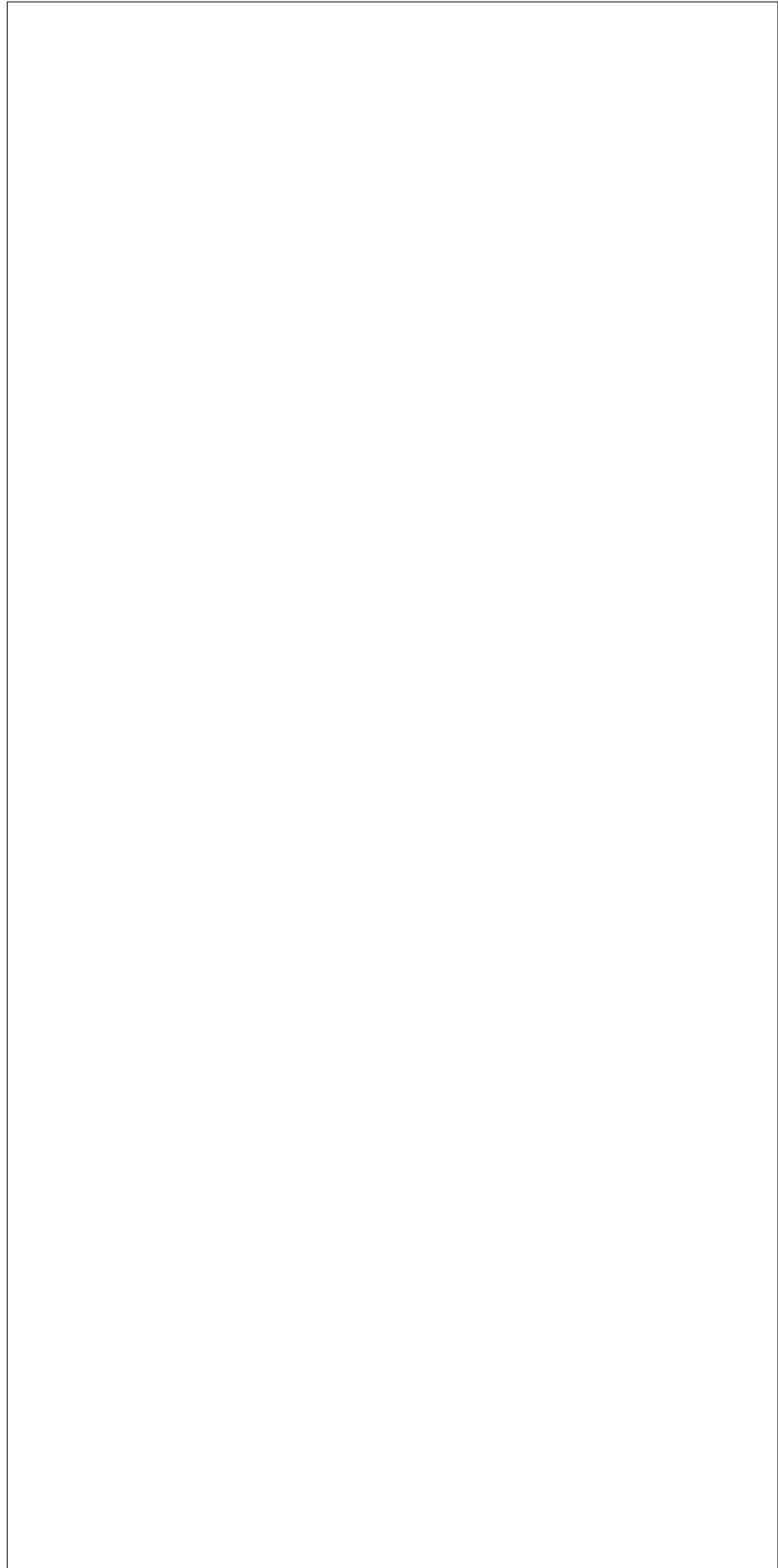
Al-
lo-
ca-
tion-
Al-
read
ists

create_L

Cre-
ate
a
list
of
BIO
Set-
ting
reco
for
a
give
node

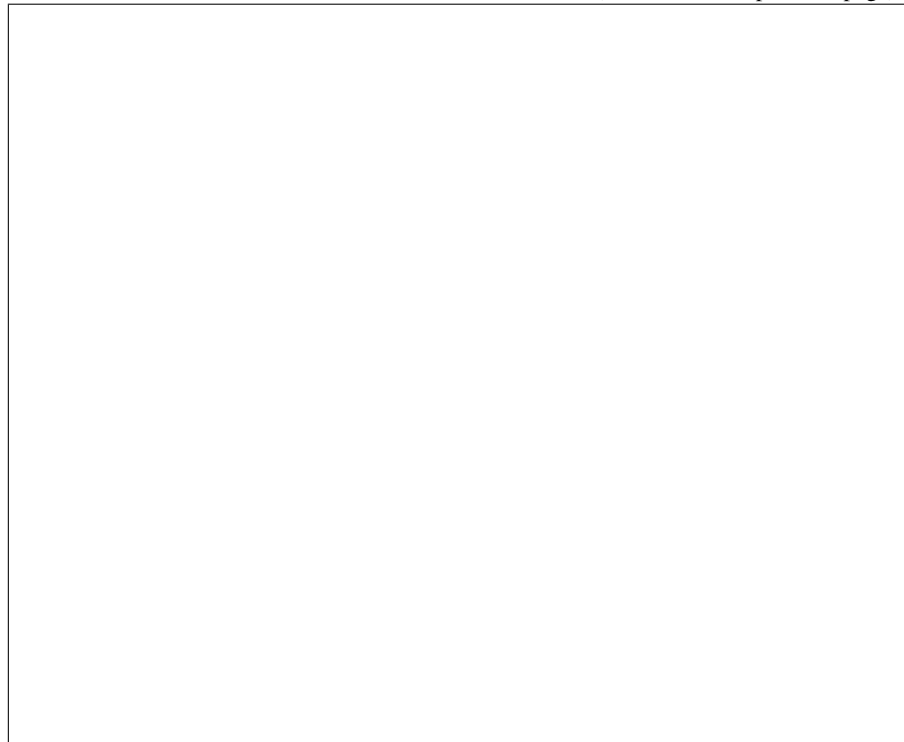
Paramete

- **node**
The
node
id.
- **set**
A
list
of
BIO
Set-
ting
to
be
cre-
ated



(continues on next page)

(continued from previous page)



-

ver:
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
BIO
Set-
ting
ob-
ject.

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

BIO
Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco
al-
read
ex-
ists.

create_

Cre-
ate
a
new
chas
sis.

Paramet

valu
Dict
of
val-
ues.

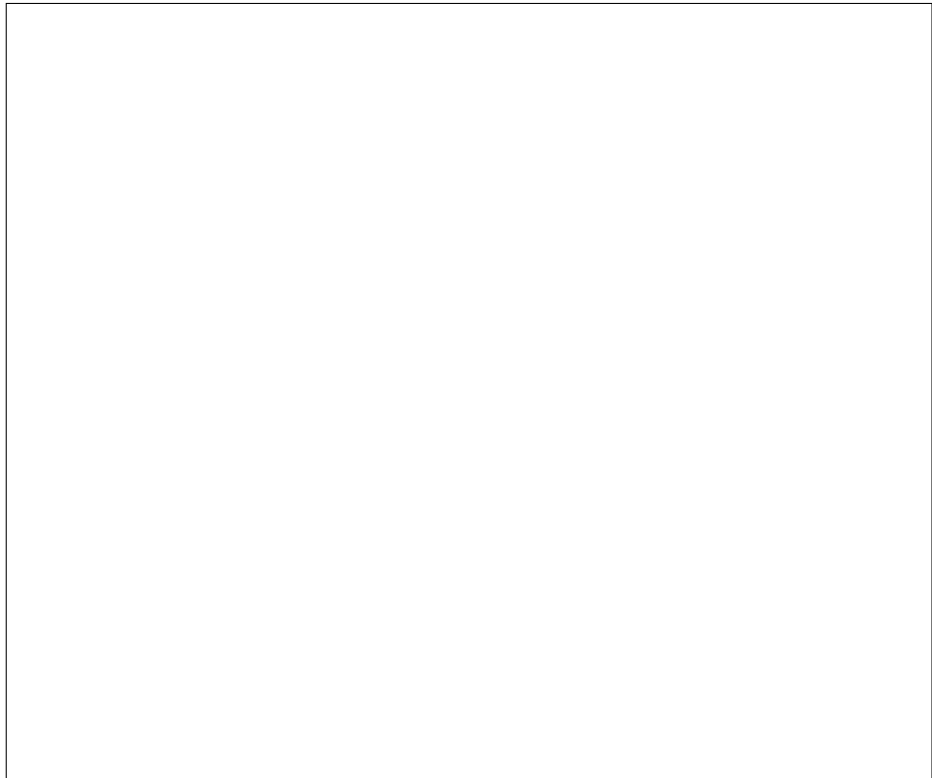
create_

Cre-
ate
a
de-
ploy
men
tem-
plate

Paramet

valu
A
dict
de-
scrib
ing
the

de-
ploy
men
tem-
plate
For
ex-
am-
ple:



Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-

ists.

Raises

DeploymentPlatformReadError if a deployment template with the same UUID exists.

Returns

A deployment template

create_1

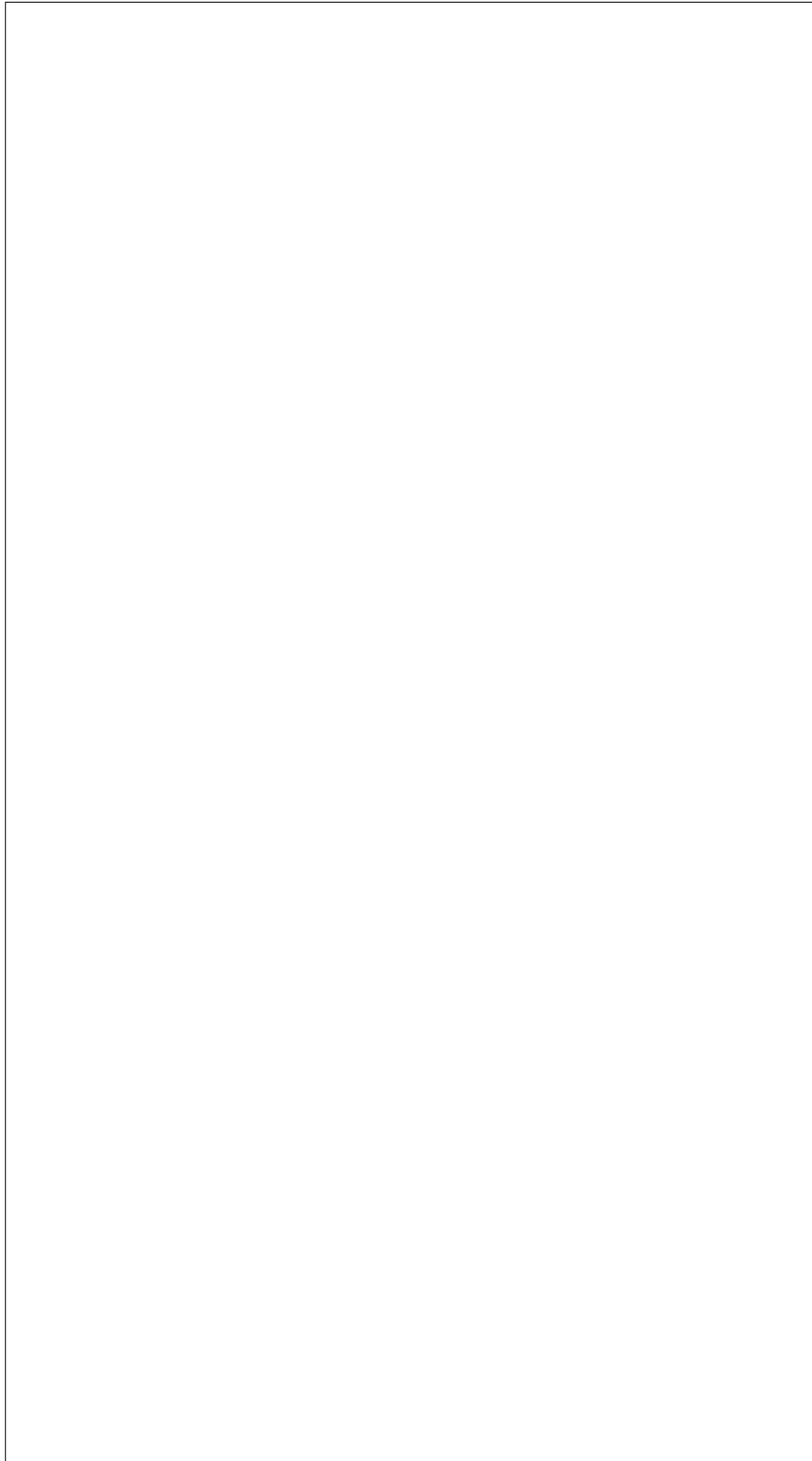
Create a new node

Parameter

A dictionary containing several items used to identify and track the

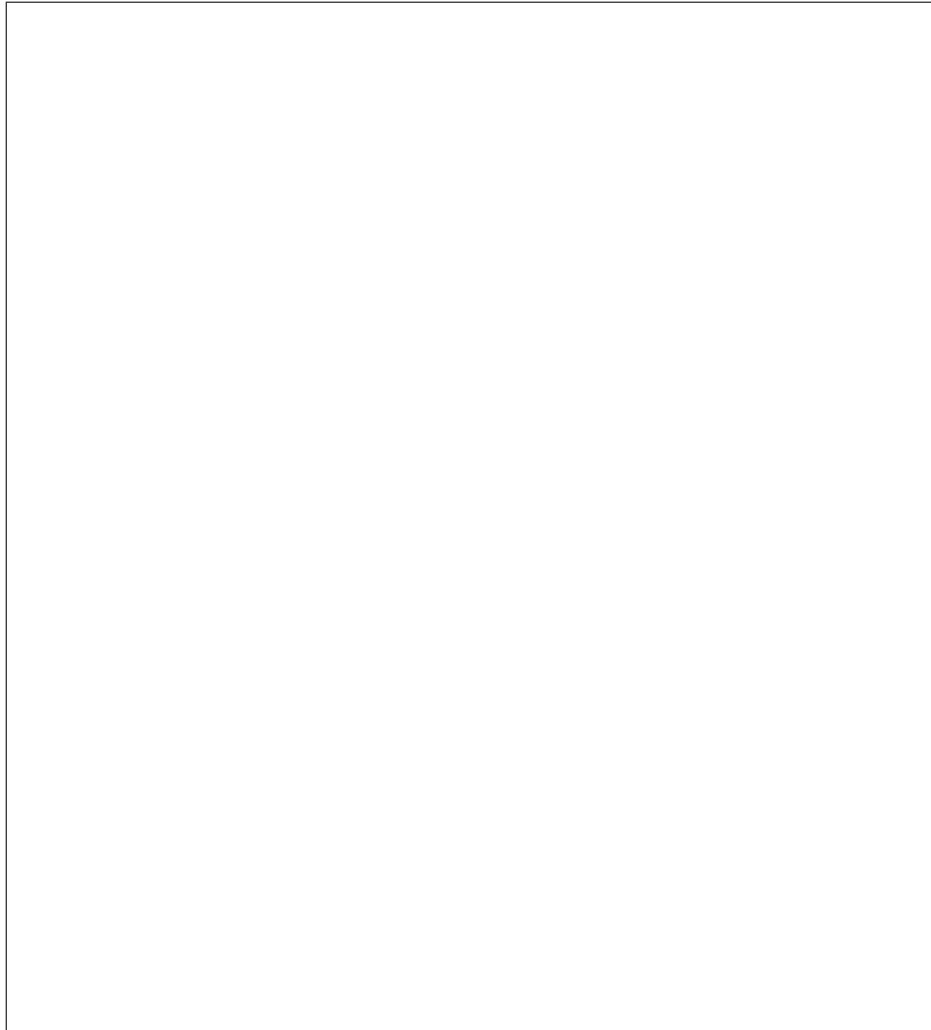
node
and
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
ues
con-
tains
tags
or
trait

Returns

A
node

create_p

Cre-
ate
a
new
port

Parameter
value
Dict
of
val-
ues.

create_p

Cre-
ate
a
new
port
group

Parameter
value
Dict
of
val-
ues
with
the
fol-
low-
ing
keys
id
uuid
name
node
ad-
dres
ex-
tra

created_at updated_at

Returns
A
port
group

Raises
Port
group
pli-

cate
Nam

Raises

Port
grou
MA
read
ists

Raises

Port
grou
read
ists

create_v

Cre-
ate
a
new
vol-
ume
con-
nec-
tor.

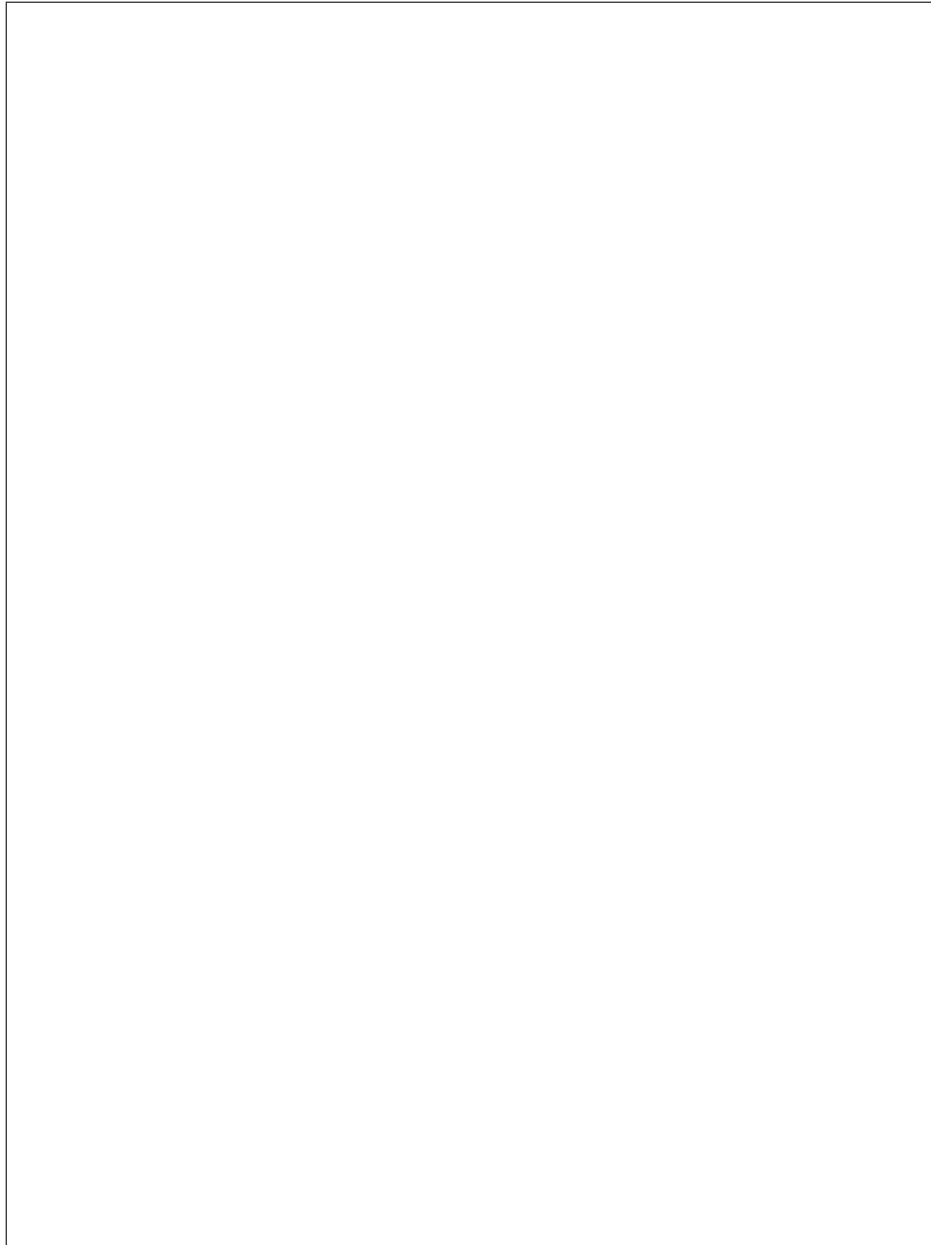
Parame

con
Dic-
tio-
nary
con-
tain-
ing
in-
for-
ma-
tion
about
the
con-
nec-
tor.
Ex-
am-
ple:



(continues on next page)

(continued from previous page)



Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torT
pe-

ists with a matching type and connector_id.

ready exists.

An-
dI-
dAl-
read
ists
If
a
con-
nec-
tor
al-
read
ex-

Raises

Vol-
ume
Con
nec-
torA
read
ists
If
a
vol-
ume
con-
nec-
tor
with
the
sam
UUI
al-

create_v

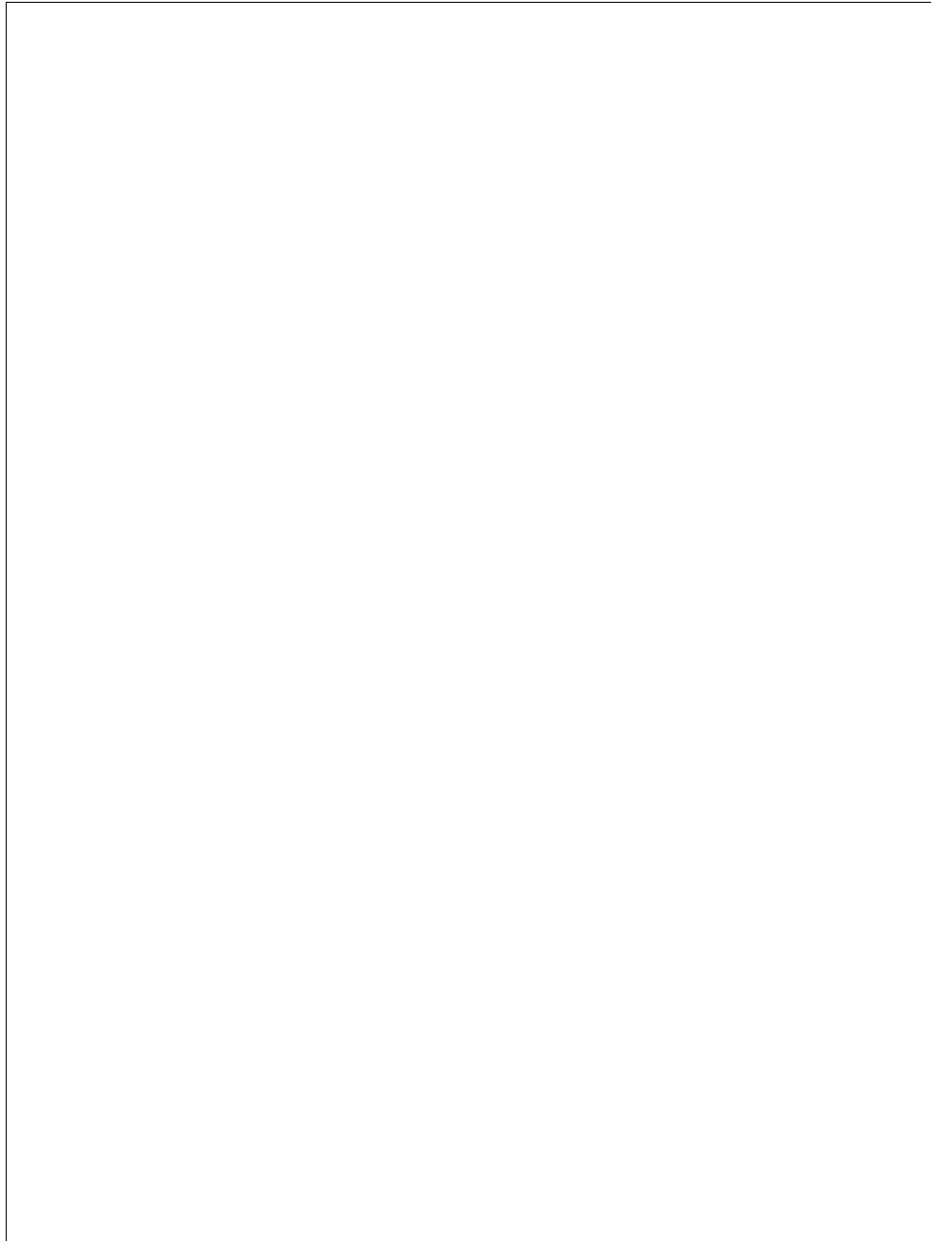
Cre-
ate
a
new
vol-
ume
tar-
get.

Paramet

tar
Dic-
tio-
nary

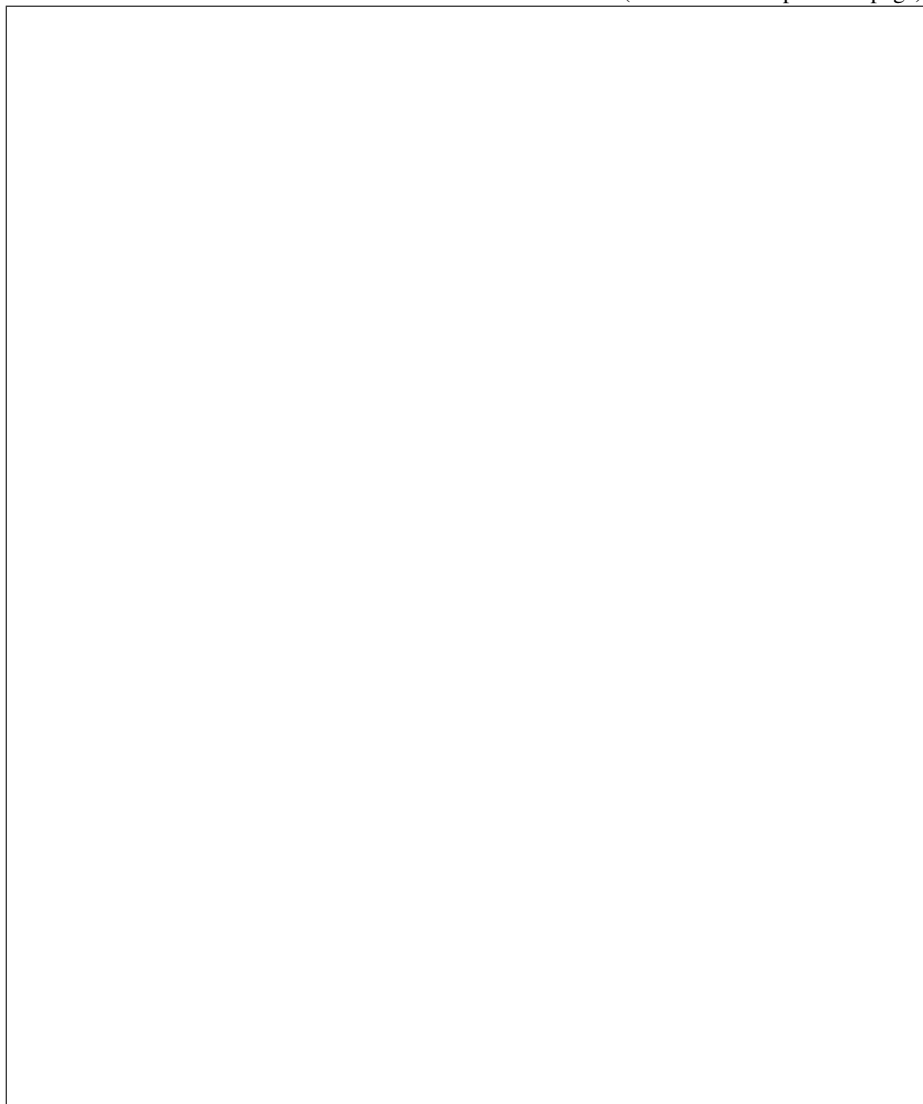
con-
tain-
ing
the
in-
for-
ma-
tion
about
the
vol-
ume
tar-
get.
Ex-

ample:



(continues on next page)

(continued from previous page)



Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-

the same boot index and node ID.

ume
tar-
get
al-
read
ex-
ists
with

Raises

Vol-
ume
ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with
the
sam
UUI
ex-
ists.

delete_**L**

Dele
a
list
of
BIO
set-
ting

Parame

- **node**
The
node
id.
- **name**
List
of
BIO

set-
ting
nam
to
be
dele

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
any
of
BIO
set-
ting
nam
is
not
foun

delete_r

Dele
spec
i-
fied
tag
from
the
node

Paramet

- **node**
The
id
of

a
node

- **tag**
A
tag
string

Raises
Nod
Not-
Four
if
the
node
is
not
foun

Raises
Nod
Tag-
Not-
Four
if
the
tag
is
not
foun

delete_r
Dele
spec
i-
fied
trait
from
the
node

Parame
• **node**
The
id
of
a
node

-

tra

A
trait
strin

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

Nod
Trai
Not-
Foun
if
the
trait
is
not
foun

destroy.

De-
stroy
an
al-
lo-
ca-
tion.

Parame

all
Al-
lo-
ca-
tion
ID
or
UUI

Raises

Al-
lo-
ca-
tion-
Not-

Four

destroy.

De-
stroy
a
chas
sis.

**Parameter
cha**

The
id
or
the
uuid
of
a
chas
sis.

destroy.

De-
stroy
a
de-
ploy
men
tem-
plate

**Parameter
temp**

ID
of
the
de-
ploy
men
tem-
plate
to
de-
stroy

Raises

De-
ploy
plate
Four
if
the

de-
ploy
tem-
plate
does
not
ex-
ist.

destroy_

De-
stroy
a
node
and
its
as-
so-
ci-
ated
re-
sour

De-
stroy
a
node
in-
clud
ing
any
as-
so-
ci-
ated
port
port
grou
tags
trait
vol-
ume
con-

nectors, and volume targets.

Parame
node
The
ID
or
UUID
of

a
node

destroy.

De-
stroy
an
port

Parame

por-
The
id
or
MA
of
a
port

destroy.

De-
stroy
a
port
grou

Parame

por-
The
UU
or
MA
of
a
port
grou

Raises

Port
grou
Notl

Raises

Port
grou
Not-
Four

destroy.

De-
stroy
a
vol-

ume
con-
nec-
tor.

Parame

iden
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
iden

does not exist.

destroy_

De-
stroy
a
vol-
ume
tar-
get.

Parameter

identifier

The UUI or in- te- ger ID of a vol- ume tar- get.

Raises

Volume get- Not- Four if a vol- ume tar- get with the spec- i- fied iden- does not

exist.

get_act:

Re- triev hard ware type for the reg- is- terec and ac-

tive
con-
duc-
tors.

Parameter use.

When
to
factor
con-
duc-
tor_
into
the
keys

Returns

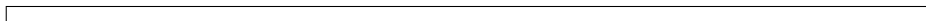
A
dict
which
map
hard
ware
type
names
to
the
set
of
host
which
sup-
port
them
For
ex-
am-

ple:



(continues on next page)

(continued from previous page)



get_all

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all
The
id
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Four

get_all

Re-
turn
an
al-
lo-
ca-
tion
rep-

re-
sen-
ta-
tion.

Parame

name

The
log-
i-
cal
name
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Foun

get_all

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all

The
uuid
of

an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Four

get_all

Re-
turn
a
list
of
al-
lo-
ca-
tion.

Paramete

- **fil:**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

node_
uuid
of
node

state
al-

lo-
ca-
tion
state

resour

re-
ques
re-
sour
class

-

lim

Max
i-
mun
num
ber
of
al-
lo-
ca-
tions
to
re-
turn

-

mar

The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

-

sor

At-
tribu
by

which
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
which
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
al-
lo-
ca-
tions

get_bios

Re-
triev
BIO
set-
ting
valu

Parame

- **node**
The
node
id.
- **name**
Strin
con-
tain-

ing
nam
of
BIO
set-
ting
to
be
re-
triev

Returns

The
BIO
Set-
ting
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
the
BIO
set-
ting
is
not
foun

get_bios

Re-
triev
BIO
set-
ting
of

a
give
node

Parameter

node
The
node
id.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
found

get_chas

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parameter

cha
The
id
of
a
chas
sis.

Returns

A
chas
sis.

get_chas

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

**Parame
cha**

The
uuid
of
a
chas
sis.

Returns

A
chas
sis.

get_chas

Re-
turn
a
list
of
chas
sis.

Parame

- **lim**
Max
i-
mun
num
ber
of
chas
sis
to

re-
turn

- **mark**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

get_conc

Re-

triev
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam
of
the
con-
duc-
tor
ser-
vice
- **onl.**
Spec
ify
the
fil-
ter
valu
on
the
*on-
line*
field
whe
quer
ing
con-
duc-
tors.
The

online field is ignored if this value is set to None.

Returns

A
con-

meet the specified online expectation.

duc-
tor.

Raises

Con
duc-
torN
Four
if
the
con-
duc-
tor
with
give
host
nam
does
not
ex-
ist
or
does

get_conc

Re-
turn
a
list
of
con-
duc-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
con-
duc-
tors
to
re-
turn

- **mark**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_dep:

Re-
triev
a
de-

ploy
men
tem-
plate
by
ID.

Parameters

template_id
ID
of
the
de-
ploy-
men-
tem-
plate
to
re-
triev

Raises

De-
ploy-
plate
Four
if
the
de-
ploy-
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy-
tem-
plate

get_deployments

Re-
triev
a
de-
ploy-
men-
tem-

plate
by
nam

Parameters

temp
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

get_dep

Re-
triev
a
de-
ploy
men
tem-
plate
by
UU

Parameter

template
URI
of
the
de-
ploy
men-
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

get_dep

Re-
triev
a
list
of
de-
ploy
men-
tem-
plate

Parameter

-

lim
Max
i-
mun
num
ber
of
de-
ploy
tem-
plate
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in

which
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
de-
ploy
tem-
plate

get_dep:

Re-
turn
a
list
of
de-
ploy
men
tem-
plate
with
one
of
a
list
of
nam

Parame

name
List
of
nam
to
fil-
ter
by.

Returns

A
list
of
de-

ploy
tem-
plate

get_node

Re-
turn
a
node

**Parameter
node**

The
id
of
a
node

Returns

A
node

get_node

Re-
turn
a
node

**Parameter
ins**

The
in-
stan-
ce
uuid
to
search
for.

Returns

A
node

Raises

In-
stan-
ceNot
Found
if
the
in-
stan-
ce
is
not

found

Raises

In-
valid
UID
if
the
in-
stan-
ce
uuid
is
in-
valid

get_node

Re-
turn
a
node

Parameter

node
The
log-
i-
cal
name
of
a
node

Returns

A
node

get_node

Find
a
node
by
any
match-
ing
port
ad-
dress

Parameter

addr
list
of
port

ad-
dres
(e.g.
MA

Returns

Nod
ob-
ject.

Raises

Nod
Not-
Four
if
non
or
sev-
eral
node
are
foun

get_node

Re-
turn
a
node

Parameter

node
The
uuid
of
a
node

Returns

A
node

get_node

Re-
turn
a
list
of
node

Parameter

•
fil

Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associa

True
|
Fals

reserv

True
|
Fals

mainte

True
|
Fals

chassis

uuid
of
chas
sis

driver

driv
nam

provis

pro-
vi-
sion
state
of
node

provis

node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-

val
in
sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
di-

rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_node

Get
node
tags
base
on
its
id.

Parameter
node

The
id
of
a
node

Returns

A
list
of
Node
Tag
ob-
jects

Raises

Node
Not-
Found
if
the
node
is
not
found

get_node

Get
node

trait
base
on
its
id.

Parameter

node
The
id
of
a
node

Returns

A
list
of
Node
Trait
ob-
jects

Raises

Node
Not-
Found
if
the
node
is
not
found

get_node

Get
spe-
cific
column
for
match-
ing
node

Re-
turn
a
list
of
the
spec-
i-
fied

ters.

colu
for
all
node
that
matc
the
spec
i-
fied
fil-

Parame

- **col:**
List
of
col-
umn
nam
to
re-
turn
De-
fault
to
id
col-
umn
whe
colu
==
Non

- **fil:**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associ:
True
|
Fals

reserv
True
|
Fals

reserv
[con
duc-
tor1
con-
duc-
tor2

maint
True
|
Fals

retirec
True
|
Fals

chassis
uuid
of
chas
sis

driver
driv
nam

provis
pro-
vi-
sion
state
of
node

provis
node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in

sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
di-
rec-
tion

in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
tu-
ples
of
the
spec
i-
fied
colu

get_not.

Re-
turn
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-
fied
ver-
sion

This
re-
turn
ob-
jects
with
ver-
sion
that
are
not

null versions (there shouldnt be any) are also returned.

the
spec
i-
fied
ver-
sion
Ob-
jects
with

Parameters

- **mod**
the
name
of
the
mod
(clas
of
de-
sired
ob-
jects
- **ver**
list
of
ver-
sion
of
ob-
jects
not
to
be
re-
turn

Returns

list
of
the
DB
ob-
jects

Raises

Iron
icEx

cep-
tion
if
there
is
no
class
as-
so-
ci-
ated
with
the
nam

get_off:

Get
a
list
con-
duc-
tors
that
are
of-
fine
(dea

**Parameter
file:**

A
field
to
re-
turn
host
nam
by
de-
fault

Returns

A
list
of
re-
ques
field
of
of-
fine
con-

duc-
tors.

get_onl:

Get
a
list
con-
duc-
tor
host
nam
that
are
on-
line
and
ac-
tive.

Returns

A
list
of
con-
duc-
tor
host
nam

get_port

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Parame

add:
The
MA
ad-
dres
of
a
port

Returns

A
port

get_port

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Paramete

por
The
id
of
a
port

Returns

A
port

get_port

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Paramete

por
The
nam
of
a
port

Returns

A
port

get_port

Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Paramete

por

The
uuid
of
a
port

Returns

A
port

get_port

Re-
turn
a
list
of
port

Paramete

-

lim

Max
i-
mun
num
ber
of
port
to
re-
turn

-

mar

the

last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_port

Re-
turn
a
net-
worl
port
grou

rep-
re-
sen-
ta-
tion.

Parame

- **add:**
The
MA
ad-
dres
of
a
port
grou

- **pro:**
A
node
own
or
lesse
to
fil-
ter
by.

Returns

A
port
grou

Raises

Port
grou
Not-
Fou

get_port

Re-
turn
a
net-
worl
port
grou
rep-
re-

sen-
ta-
tion.

Parame

por

The
id
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Four

get_port

Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

nam

The
log-
i-
cal
nam
of
a
port
grou

Returns

A
port

grou

Raises

Port

grou

Not-

Four

get_port

Re-

turn

a

net-

work

port

grou

rep-

re-

sen-

ta-

tion.

Paramete

por

The

uuid

of

a

port

grou

Returns

A

port

grou

Raises

Port

grou

Not-

Four

get_port

Re-

turn

a

list

of

port

grou

Paramete

- **limit**
Max-
i-
mun-
num-
ber
of
port-
grou-
to
re-
turn
- **mark**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sort**
At-
tribu-
by
whic-
re-
sults
shou-
be
sorte
- **sort**
Di-
rec-
tion
in
whic

re-
sults
shou
be
sorte
(asc
desc

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns
A
list
of
port
grou

get_port

List
all
the
port
grou
for
a
give
node

Paramete

- **node**
The
in-
te-
ger
node
ID.

- **lim**
Max

i-
mun
num
ber
of
port
grou
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in
whic
re-
sults
shou

be
sorted
(asc.
desc

- **pro.**
A
node
own
or
less
to
fil-
ter
by.

Returns

A
list
of
port
group

get_port

List
all
the
port
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber

of
port
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sor**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A list of ports

get_ports

List all the ports for a given port group

Parameters

- **port**
The integer port group ID.
- **limit**
Maximum number of ports to return
- **marker**
The last item of the previous

page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
port

get_volu
Re-
turn
a
vol-
ume
con-
nec-
tor

rep-
re-
sen-
ta-
tion.

Parameters

db_
The
in-
te-
ger
data
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec-
i-
fied
ID.

Raises

Vol-
ume
Con-
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor
with
the
spec

is not found.

i-
fied
ID

get_volu

Re-
turn
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

con
The
UUID
of
a
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UUID

Raises

Vol-
ume
Con-
nec-
torN
Four
If

is not found.

a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UUI

get_volu

Re-
turn
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-
turn
- **mar**
The
last
item
of

the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a list of Volume objects

Returns

A list of volume connectors.

Raises

InvalidParameterException
ValueError
If sort does not exist.

get_volumes

List all the volume connectors for a given node

Parameters

- **node**
The

in-
te-
ger
node
ID.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be

sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of
Vol1
ob-
jects

Returns
A
list
of
vol-
ume
con-
nec-
tors.

Raises
In-
valid
Pa-
ram-
e-

ter-
Valu
If
sort_
does
not
ex-
ist.

get_volu

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

db_
The
data
pri-
mar-
key
(in-
te-
ger)
ID
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-

Not-
Four
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists.

get_volu

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

uui
The
UUI
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four

if
no
vol-
ume
tar-
get
with
this
UUID
ex-
ists.

get_volumes

Re-
turn
a
list
of
vol-
ume
tar-
gets.

Parameters

- **limit**
Max-
i-
mun-
num-
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **marker**
the
last
item
of
the
pre-
vi-
ous

page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list

of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

get_volu

List
all
the
vol-
ume
tar-
gets
for
a
give
node

Parame

- **node**
The
in-
te-
ger
node
ID.

- **limit**
Maximum number of volume targets to return

- **marker**
The last item of the previous page we return the next result set.

- **sortBy**
Attribute by which results should be sorted

- **sortDir**
Direction

in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-

ist.

get_volumes

List all the volumes targeted for a given volume id.

Parameters

- **volume_id**
The UUID of the volume.
- **limit**
Maximum number of volumes targeted to return.
- **marker**
The last item of the previous

vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
vol-
ume
tar-
gets.

Raises
In-
valid
Pa-
ram-

e-
ter-
Valu
if
sort
does
not
ex-
ist.

list_con

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor.

Returns

List
of
Con
ob-
jects

list_ha

List
reg-
is-
tere
hard
ware
in-
ter-

filter by. :returns: list of `ConductorHardwareInterfaces` objects.

face
for
give
hard
ware
type
This
is
re-
stric
to
only
ac-
tive
con-
duc-
tors.
:para
hard
ware
list
of
hard
ware
type
to

node_tag

Che
if
the
spec
i-
fied
tag
ex-
ist
on
the
node

Parameter

- **node**
The
id
of
a

node

- **tag**
A
tag
string

Returns

True
if
the
tag
ex-
ists
oth-
er-
wise
False

Raises

Node
Not-
Found
if
the
node
is
not
found

node_traits

Check
if
the
spec-
ified
trait
ex-
ists
on
the
node

Parameters

- **node**
The
id
of

a
node

- **tra**
A
trait
strin

Returns

True
if
the
trait
ex-
ists
oth-
er-
wise
Fals

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

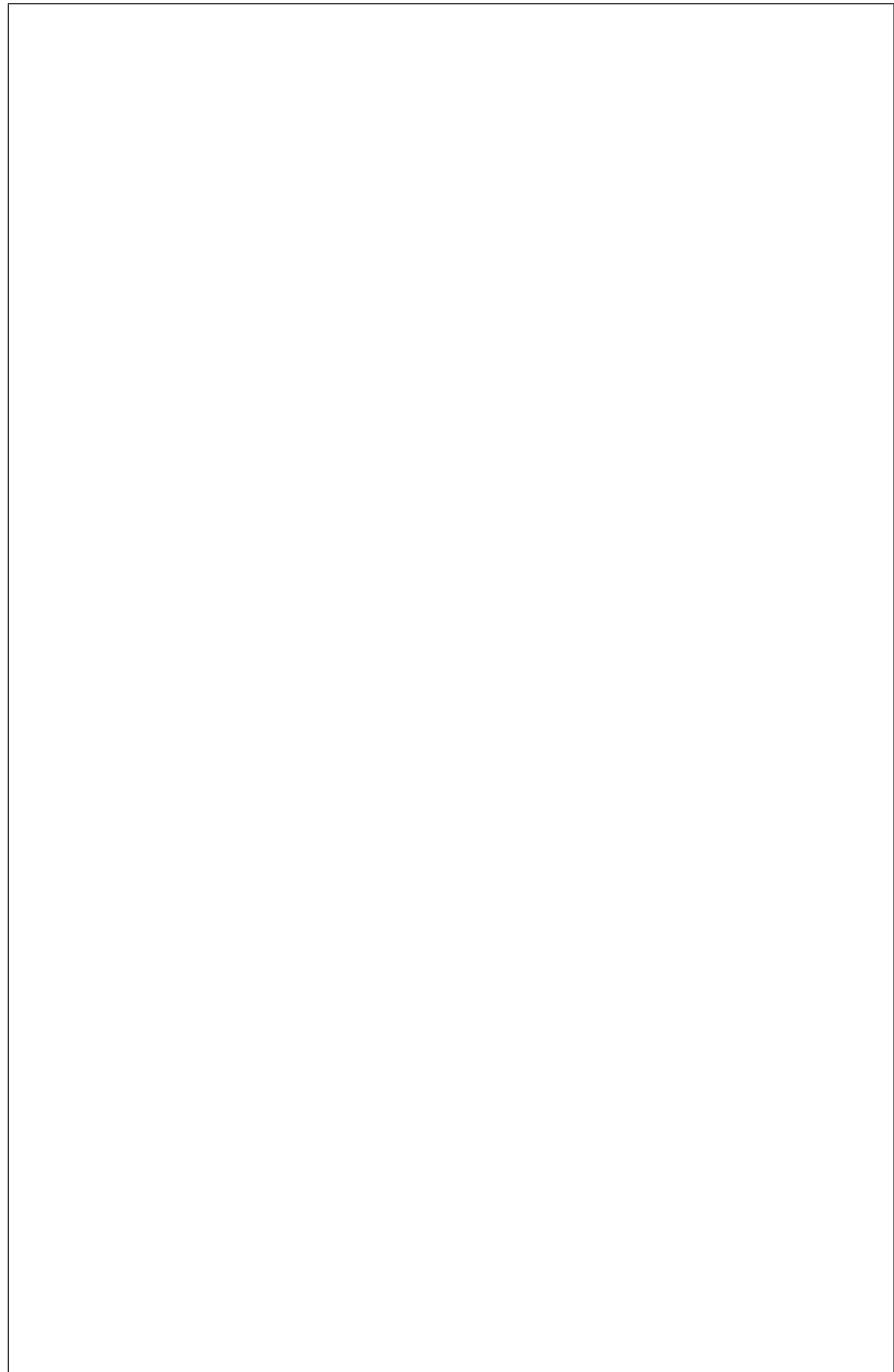
register

Reg-
is-
ter
an
ac-
tive
con-
duc-
tor
with
the
clus
ter.

Paramet

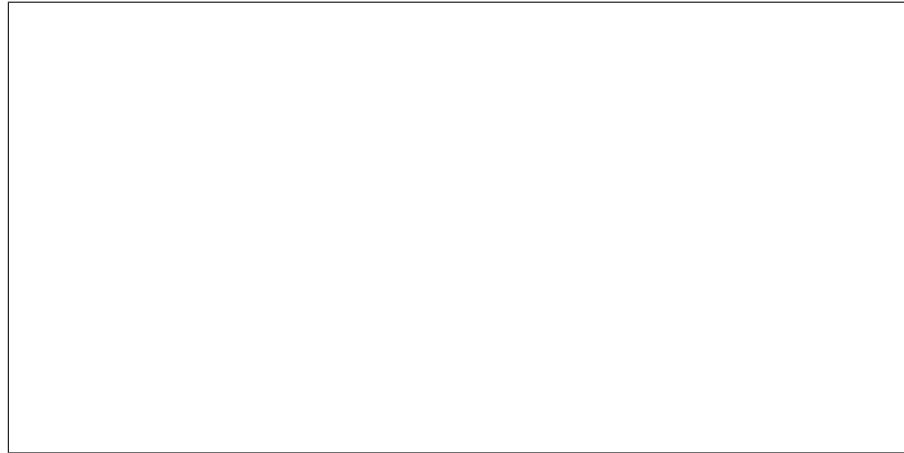
- **val**
A

dict
of
val-
ues
whic
mus
con-
tain
the
fol-
low-
ing:



(continues on next page)

(continued from previous page)



line record is found. When true, will overwrite the existing record. Default: False.

- **update**
When
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion
when
a
con-
flict-
ing
on-

Returns

A
con-
duc-
tor.

Raises

Con-
duc-
torA
read
is-
tere-

register

Reg

is-
ters
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

- **con**
Data
ID
of
con-
duc-
tor
to
reg-
is-
ter
for.
- **har**
Nam
of
hard
ware
type
for
the
in-
ter-
face
- **int**
Type
of
in-
ter-
face
e.g.
de-
ploy

or
boot

- **int**
List
of
in-
ter-
face
nam
to
reg-
is-
ter.

- **def**
Strin
the
de-
fault
in-
ter-
face
for
this
hard
ware
type
and
in-
ter-
face
type

Raises
Con
duc-
torH
ware
ter-
face
sAl-
read
is-
tere
if
at
least
one
of
the

in the combination of all parameters is already registered.

in-
ter-
face

release_

Re-
lease
the
rese
va-
tion
on
a
node

Paramete

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **node**
A
node
id
or
uuid

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
re-
serv
by
an-
othe
host

Raises

Nod
Not-
Lock
if
the
node
was
foun
to
not
have
a
rese
va-
tion
at
all.

reserve_

Re-
serv
a
node

To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing

performed, mark it reserved by this host.

the
give
Nod
whil
a
Task
is

Parame

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**
A
node
id
or
uuid

Returns

A
Nod
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod

Lock
if
the
node
is
al-
read
re-
serv

set_node

Re-
plac
all
of
the
node
tags
with
spec
i-
fied
list
of
tags

This
ig-
nore
du-
pli-
cate
tags
in
the
spec
i-
fied
list.

Paramete

- **node**
The
id
of
a
node
- **tag**

List
of
tags

Returns

A
list
of
Node
Tag
ob-
jects

Raises

Node
Not-
Found
if
the
node
is
not
found

set_node

Re-
plac
all
of
the
node
traits
with
spec
i-
fied
list
of
traits

This
ig-
nore
du-
pli-
cate
traits
in
the
spec
i-
fied

list.

Parameters

- **node**
The id of a node
- **traits**
List of traits
- **version**
the version of the object.

Returns

A list of NodeTrait objects

Raises

InvalidParameterError
ValueError
if setting the trait would exceed

limit.

thus guarding against races.

the
per-
node
trait

Raises

Nod
Not-
Four
if
the
node
is
not
foun

take_ove

Do
a
take
over
for
an
al-
lo-
ca-
tion.

The
al-
lo-
ca-
tion
is
only
up-
date
if
the
old
con-
duc-
tor
matc
the
pro-
vide
valu

Parame

cation.

- **all**
Al-
lo-
ca-
tion
ID

- **old**
The
con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
rent
con-
of
the
al-
lo-

- **new**
The
con-
duc-
tor
ID
of
the
new
con-

Returns

True
if
the
take
over
was
suc-
cess

ful,
Fals
oth-
er-
wise

Raises

Al-
lo-
ca-
tion-
Not-
Four

touch_co

Mar
a
con-
duc-
tor
as
ac-
tive
by
up-
dat-
ing
its
up-
date
prop
erty.

Parame

hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

touch_no

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning
Mar
the
node
pro-
vi-
sion
ing
as
run-
ning
by
up-
dat-
ing
its
pro-
vi-
sion
prop
erty.

Parame
node
The
id
of
a
node

Raises
Nod
Not-
Four

unregist
Re-
mov
this
con-
duc-
tor

from
the
ser-
vice
reg-
istry
im-
me-
di-
ately

Parameter
hostname

The
hostname
name
of
this
con-
duc-
tor
ser-
vice

Raises

Con-
duc-
torN
Four

unregister

Un-
reg-
is-
ters
all
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parameter
conducting

Data
ID
of

con-
duc-
tor
to
un-
reg-
is-
ter
for.

unset_no

Re-
mov
all
tags
of
the
node

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

unset_no

Re-
mov
all
trait
of
the
node

Parame

nod
The
id
of

a
node

Raises

Node
Not-
Found
if
the
node
is
not
found

update_a

Up-
date
prop
er-
ties
of
an
al-
lo-
ca-
tion.

Paramete

- **all**
Al-
lo-
ca-
tion
ID
- **valu**
Dict
of
val-
ues
to
up-
date
- **upd**
If
True
and

allocation

node
is
up-
date
up-
date
the
node
with
in-
stan-
and
trait
from
the

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Four

Raises

Al-
lo-
ca-
tion
pli-
cate
Nam

Raises

In-
stan-
As-
so-
ci-
ated

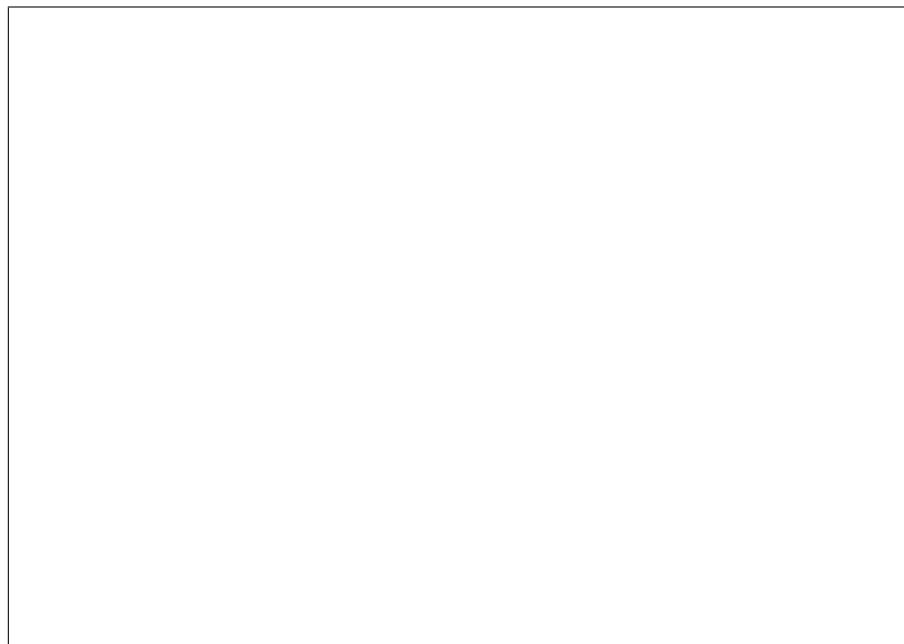
Raises

Nod
As-
so-
ci-

ated
update_L
Up-
date
a
list
of
BIO
Set-
ting
reco

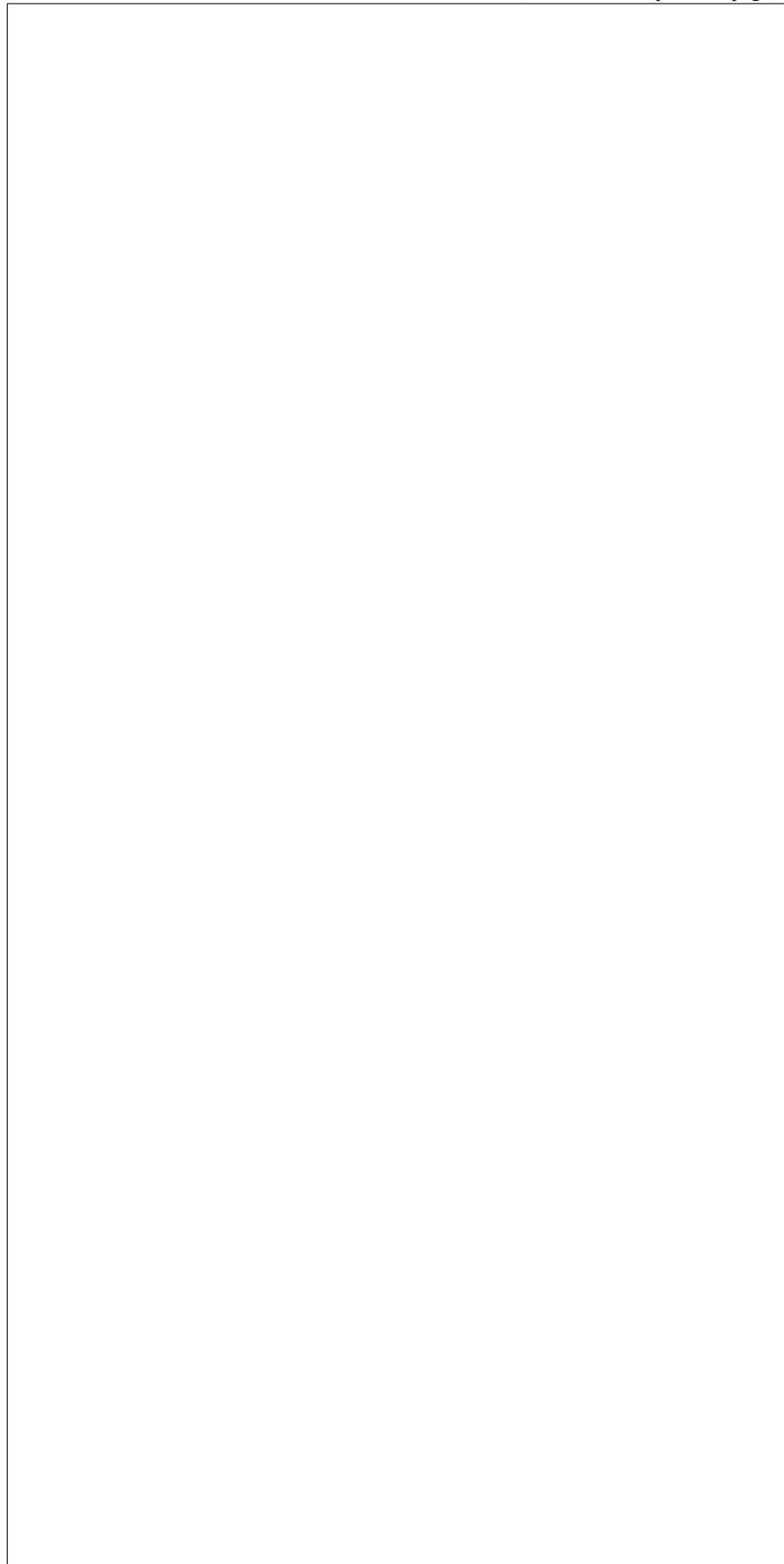
Parame

- **node**
The
node
id.
- **set:**
A
list
of
BIO
Set-
tings
to
be
up-
date



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



- **ver:**
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

BIO
Set-
ting
Not-
Foun
if
any
of
the
set-
tings
is
not
foun

update_

Up-

date
prop
er-
ties
of
an
chas
sis.

Parame

- **cha**
The
id
or
the
uuid
of
a
chas
sis.

- **valu**
Dict
of
val-
ues
to
up-
date

Returns

A
chas
sis.

update_c

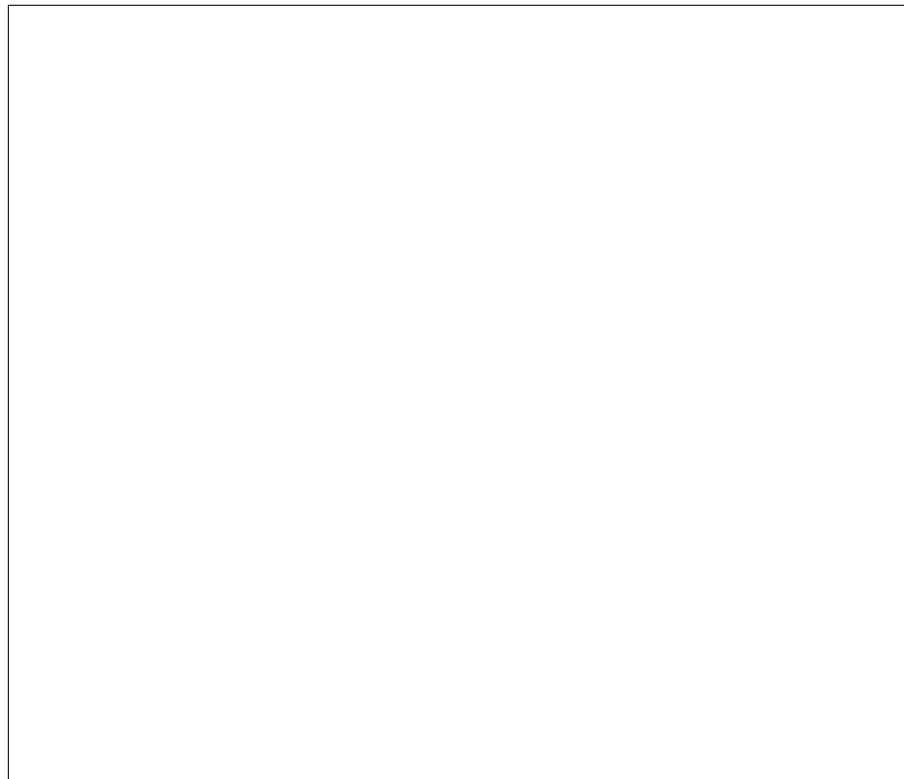
Up-
date
a
de-
ploy
men
tem-
plate

Parame

- **temp**

ID
of
the
de-
ploy
men
tem-
plate
to
up-
date

- **value**
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

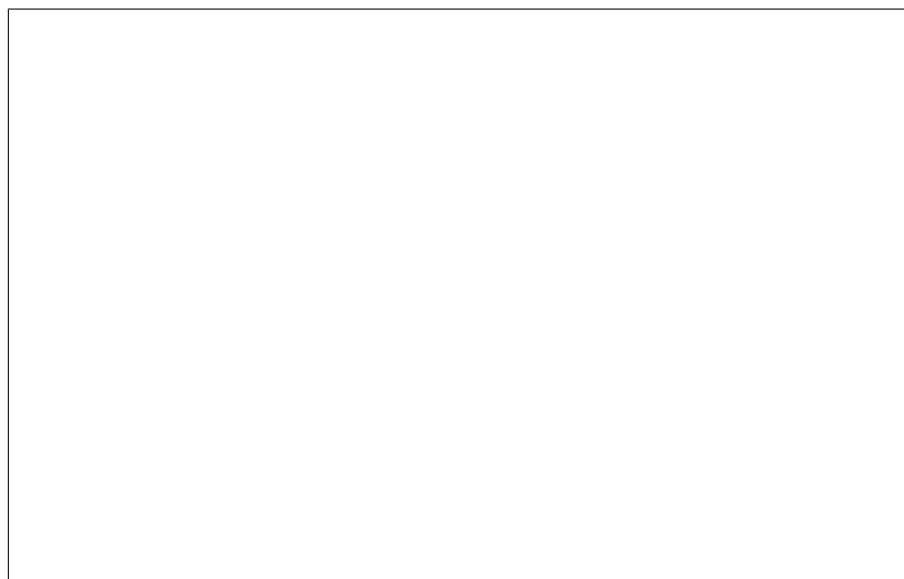
update_1

Up-
date
prop
er-
ties
of
a

node
Parameter

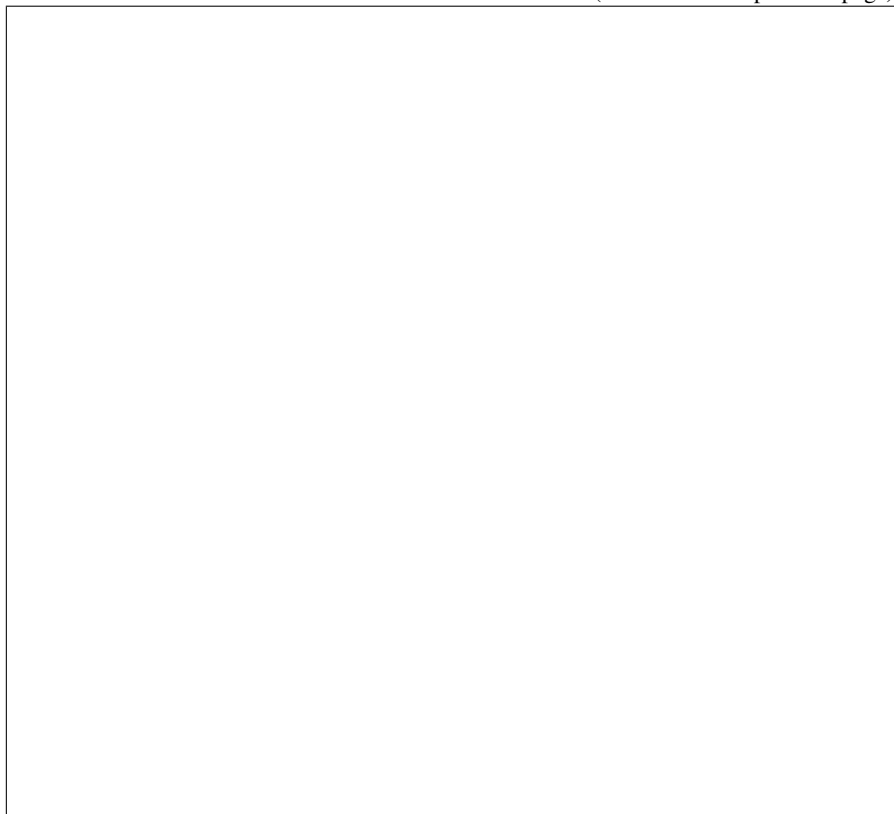
- **node**
The id or uuid of a node
- **value**
Dictionary of values to update. May be a partial list, eg. when setting the

properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns

A
node

Raises

Nod
As-
so-
ci-
ated

Raises

Nod
Not-
Four

update_p

Up-
date
prop
er-
ties
of
an
port

Parame

- **por**
The
id
or
MA
of
a
port

- **val**
Dict
of
val-
ues
to
up-
date

Returns
A
port

update_p
Up-
date
prop
er-
ties
of
a
port
grou

Paramet

- **por**
The
UI
or
MA
of
a
port
grou

- **val**
Dict
of

address extra created_at updated_at

val-
ues
to
up-
date
May
con-
tain
the
fol-
low-
ing
keys
uuid
nam
node

Returns

A
port
group

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

Port
group
Not-
Four

Raises

Port
group
pli-
cate
Nam

Raises

Port
group
MA
read
ists

update_t

dates them to that version.

Up-
date
ob-
jects
to
their
lat-
est
know
ver-
sion

This
scan
all
the
ta-
bles
and
for
ob-
jects
that
are
not
in
their
lat-
est
ver-
sion
up-

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num

all the objects will be migrated.

the beginning of this call) and 2. the number of migrated objects.

ber
of
ob-
jects
to
mi-
grate
Mus
be
>=
0.
If
zero

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that
need
to
be
mi-
grate
(at

update_v

Up-
date
prop
er-
ties
of
a
vol-
ume
con-
nec-
tor.

Parame

•

ides
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-
tor.

•

conu
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
con-
nec-
tor
to
up-
date

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con-
nec-

exists with a matching type and connector_id field.

does not exist.

torT
pe-
An-
dI-
dAl-
read
ists
If
an-
othe
con-
nec-
tor
al-
read

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
iden

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
Whe
a
UUI
is
in-

clud
in
con-
nec-
tor_

update_v

Up-
date
in-
for-
ma-
tion
for
a
vol-
ume
tar-
get.

Parame

- **iden**
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.
- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion

date.

about
vol-
ume
tar-
get
to
up-

Returns

A
vol-
ume
tar-
get.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
UI
is
in-
clud
in
tar-
get_

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists

the same boot index and node ID.

with

Raises

Vol-

ume

get-

Not-

Four

if

no

vol-

ume

tar-

get

with

this

iden

ex-

ists.

ironic.d

ironic.d

ironic.d

Add

an

iden

tity

fil-

ter

to

a

quer

Fil-

ters

re-

sults

by

ID,

if

sup-

plied

valu

is

a

valid

in-

te-

ger.

tempts to filter results by UUID.

Oth-
er-
wise
at-

Parameter

- **query:**
Initial query to add filter to.
- **value:**
Value for filtering results by.

Returns

Modified query

`ironic.d`

`ironic.d`

Add a port specification filter to a query

Filters results

attempts to filter results by identity.

sults
by
ad-
dres
if
sup-
plie
valu
is
a
vali
MA
ad-
dres
Oth-
er-
wise

Paramet

- **que.**
Ini-
tial
quer
to
add
fil-
ter
to.
- **valu**
Valu
for
fil-
ter-
ing
re-
sults
by.

Returns

Mod
i-
fied
quer

ironic.d

ironic.d

ironic.d

ironic.d

ironic.d

Add

a

port

spec

fil-

ter

to

a

quer

Fil-

ters

re-

sults

by

ad-

dres

if

sup-

pliee

valu

is

a

valid

MA

ad-

dres

Oth-

er-

wise

attempts to filter results by identity.

Paramet

-

que:

Ini-

tial

quer

to

add

fil-

ter

to.

-

valu
Valu
for
fil-
ter-
ing
re-
sults
by.

Returns

Mod
i-
fied
quer

ironic.d

ironic.d

ironic.d

ironic.d

ironic.d

The
back
end
is
this
mod
ule
it-
self.

ironic.d

Que
help
for
sim-
pler
ses-
sion
us-
age.

Paramet

ses
if
pres
the

ses-
sion
to
use

ironic.db.sqlalchemy.migration module

ironic.d

Cre-
ate
data
sche
from
mod
els
de-
scrip
tion.

Can
be
used
for
ini-
tial
in-
stal-
la-
tion
in-
stead
of
up-
grad

ironic.d

Use
for
dow
grad
ing
data

Paramet

ver:
(str)
De-
sirec
data
ver-

sion
ironic.d

Cre-
ates
tem-
plate
for
mi-
gra-
tion.

Paramet

- **mes**
(*str*)
Text
that
will
be
used
for
mi-
gra-
tion
ti-
tle

- **aut**
(*boo*)
If
True
-

gen-
er-
ates
diff
base
on
cur-
rent
data
state

ironic.d
Stan
data
with

cent revision

pro-
vide
re-
vi-
sion
Don
run
any
mi-
gra-
tions

Paramet
rev:
(*str*)
Sho
mat
one
from
repo
i-
tory
or
head
-

to
stan
data
with
mos
re-

ironic.d

Use
for
up-
grad
ing
data

Paramet
ver:
(*str*)
De-
sirec
data
ver-
sion

ironic.d

ironic.db.sqlalchemy.models module

Cur-
rent
data
ver-
sion

Returns
Data
ver-
sion

Return t
strin

SQL
mod
els
for
bare
data

class ir

Base
sql
orm
dec
Base

Rep
re-
sent
an
al-
lo-
ca-
tion
of
a
node
for
de-
ploy
men

candidat

conduct

created_

extra

id

last_err

name

node_id

owner

resource

state

traits

updated_

uuid

version

class ir

Base

sql

orm

dec

Bas

Rep

re-

sent

a

bios

set-

ting

of

a

bare
meta
node

allowabl
attribu
created
lower_bo
max_leng
min_leng
name
node_id
read_on
reset_re
unique
updated
upper_bo
value
version

class ir
Base
sql.
orm
dec
Base
Rep
re-

sent
a
hard
ware
chas
sis.

created

descript

extra

id

updated

uuid

version

class ir

Base
sql
orm
dec
Base

Rep
re-
sent
a
con-
duc-
tor
ser-
vice
en-
try.

conduct

created

drivers

hostname

id

online

updated_

version

class ir

Base
sql
orm
dec
Bas

In-
ter-
nal
ta-
ble
used
to
track
what
is
load
on
each
con-
duc-
tor.

conducto

created_

default

hardware

id

interfa

interfa

updated_

version

class ir

Base

sql

orm

dec

Base

Rep

re-

sent

a

de-

ploy

men

tem-

plate

created_

extra

id

name

updated_

uuid

version

class ir

Base

sql

orm
dec:
Bas
Rep
re-
sent
a
de-
ploy
men
step
in
a
de-
ploy
men
tem-
plate

args

created_

deploy_t

deploy_t

id

interfa

priority

step

updated_

version

class ir

Base
osl
sql
mod
Tim

osl
sql
mod
Mod

as_dict

metadata

version

class ir

Base
sql
orm
dec
Base
Rep
re-
sent
a
bare
meta
node

allocat

automat

bios_int

boot_int

chassis

clean_s

conduct

conduct

console

console

created

deploy_

deploy_s

descript

driver

driver_

driver_

extra

fault

id

inspect

inspect:

inspect:

instanc

instanc

last_err

lessee

maintena

maintena

managem

name

network

network

owner

power_in

power_st

propert:

protect

protect

provisio

provisio

raid_con

raid_int

rescue_:

reservat

resourc

retired

retired

storage

target_p

target_p

target_p

updated_

uuid

vendor_i

version

class ir

Base

sql

orm

dec

Base

Rep

re-

sent

a

tag

of

a

bare

meta

node

created_

node

node_id

tag

updated_

version

class ir

Base
sql
orm
dec
Bas

Rep
re-
sent
a
trait
of
a
bare
meta
node

created_

node

node_id

trait

updated_

version

class ir

Base
sql
orm
dec
Bas

Rep
re-
sent
a
net-
worl
port
of
a
bare

meta
node

address

created

extra

id

internal

is_smart

local_l

name

node_id

physical

portgro

pxe_enab

updated

uuid

version

class ir

Base
sql
orm
dec
Base

Rep
re-
sent

a
group
of
net-
work
port
of
a
bare
meta
node

address

created_

extra

id

internal

mode

name

node_id

propert

standal

updated_

uuid

version

class ir

Base
sql.
orm

dec:
Bas
Rep
re-
sent
a
vol-
ume
con-
nec-
tor
of
a
bare
meta
node

connect

created_

extra

id

node_id

type

updated_

uuid

version

class ir

Base
sql
orm
dec:
Bas
Rep
re-
sent

a
vol-
ume
tar-
get
of
a
bare
meta
node

boot_ino

created_

extra

id

node_id

propert

updated_

uuid

version

volume_

volume_

ironic.d

Re-
turn
the
mod
class
with
the
spec
i-
fied

nam
Parameter
mod
the
nam
of
the
class
Returns
the
class
with
the
spec
i-
fied
nam
Raises
Ex-
cep-
tion
if
there
is
no
class
as-
so-
ci-
ated
with
the
nam

ironic.d

Module contents

Submodules

ironic.db.api module

Base
class
for
stor-
age

en-
gine

class ir

Base
obj

Base
class
for
stor-
age
sys-
tem
con-
nec-
tion

abstract

Add
tag
to
the
node

If
the
node
and
tag
pair
al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parame

- **nod**
The
id
of
a
node

-

tag

A tag string

Returns

the NodeTag object.

Raises

NodeNotFoundError if the node is not found

abstract

Add trait to the node

If the node and trait pair already exists, this should still succeed

Parameter

-

node

The

id
of
a
node

- **tra**
A
trait
strin

- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
the
Nod
Trai
ob-
ject.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises
Nod
Not-
Four
if

the
node
is
not
found

abstract

Check
a
list
of
node
iden-
ti-
ties
and
map
it
to
UUIDs

This
call
takes
a
list
of
node
names
and/
UUIDs
and
tries
to
con-
vert
them
to
UUIDs
It
fails

early if any identities cannot possibly be used as names or UUIDs.

Parameters

identities
List
of
iden-
ti-
ties.

Returns

A
map
ping
from
re-
ques
iden
ti-
ties
to
node
UU

Raises

Nod
Not-
Four
if
som
iden
ti-
ties
were
not
foun
or
can-
not
be
valid
nam
or
UU

abstract

Che
the
who
data
for
in-
com
pat-
i-
ble
ob-
jects

This
scan
all
the

not specified in *ironic.common.release_mappings.RELEASE_MAPPING*.

ta-
bles
in
sear
of
ob-
jects
that
are
not
sup-
port
i.e.,
thos
that
are

Parame

ign

List
of
mod
nam
to
skip

Returns

A
Boo
True
if
all
the
ob-
jects
have
sup-
port
ver-
sion
Fals
oth-
er-
wise

abstract

Cre-
ate
a
new
al-

lo-
ca-
tion.

Parameters
value
Dict
of
val-
ues
to
cre-
ate
an
al-
lo-
ca-
tion
with

Returns
An
al-
lo-
ca-
tion

Raises
Al-
lo-
ca-
tion
pli-
cate
Nam

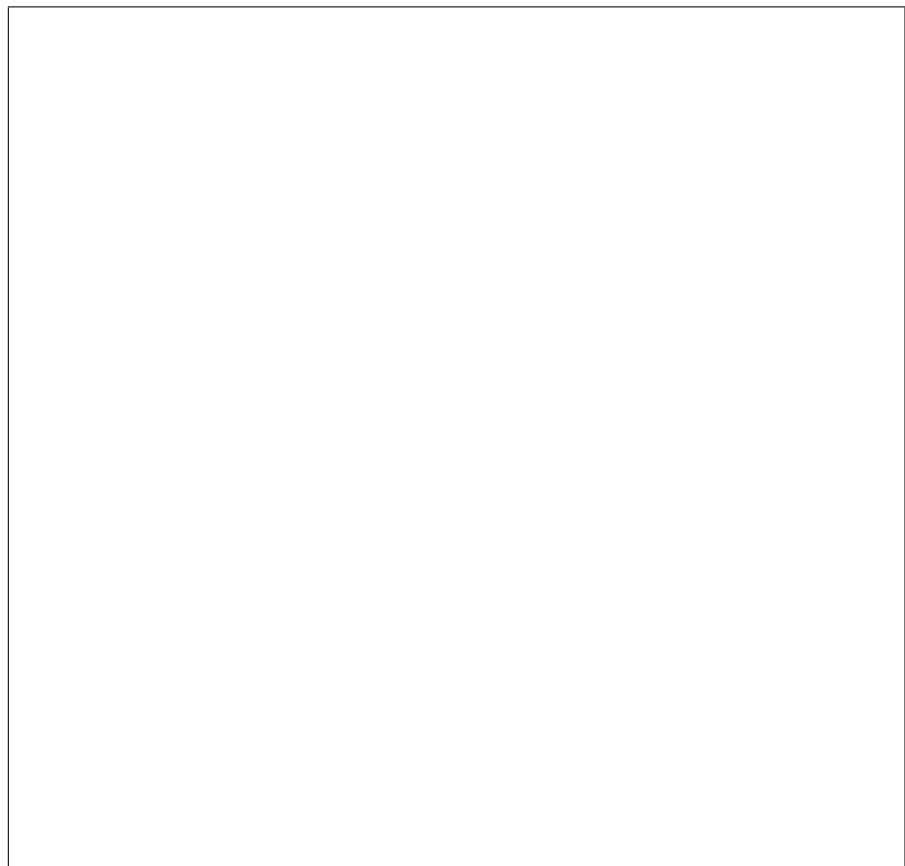
Raises
Al-
lo-
ca-
tion-
Al-
read
ists

abstract
Cre-
ate
a
list
of
BIO
Set-

ting
reco
for
a
give
node

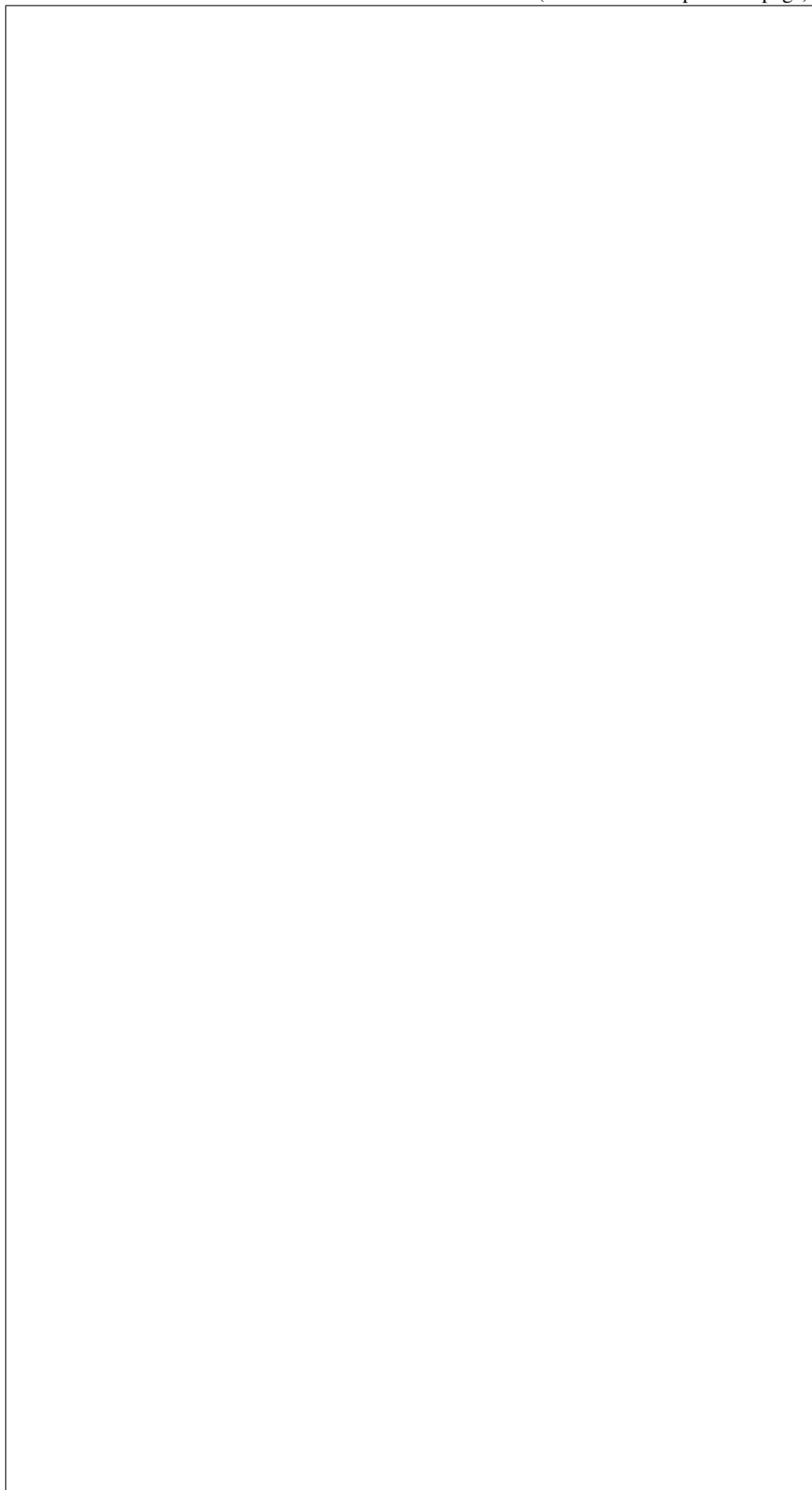
Parame

- **node**
The
node
id.
- **set:**
A
list
of
BIO
Set-
tings
to
be
cre-
ated



(continues on next page)

(continued from previous page)



•
ver:
the
ver-

sion
of
the
ob-
ject.

Returns

A
list
of
BIO
Set-
ting
ob-
ject.

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

BIO
Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco
al-
read
ex-
ists.

abstract

Cre-
ate
a
new
chas

sis.

Parameter

value

Dict

of

val-

ues.

abstract

Cre-

ate

a

de-

ploy

men

tem-

plate

Parameter

value

A

dict

de-

scrib

ing

the

de-

ploy

men

tem-

plate

For

ex-

am-

ple:



(continues on next page)

(continued from previous page)



Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

De-
ploy
plate
read
ists
if
a
de-
ploy
tem-
plate
with
the
sam
UUI
ex-
ists.

Returns

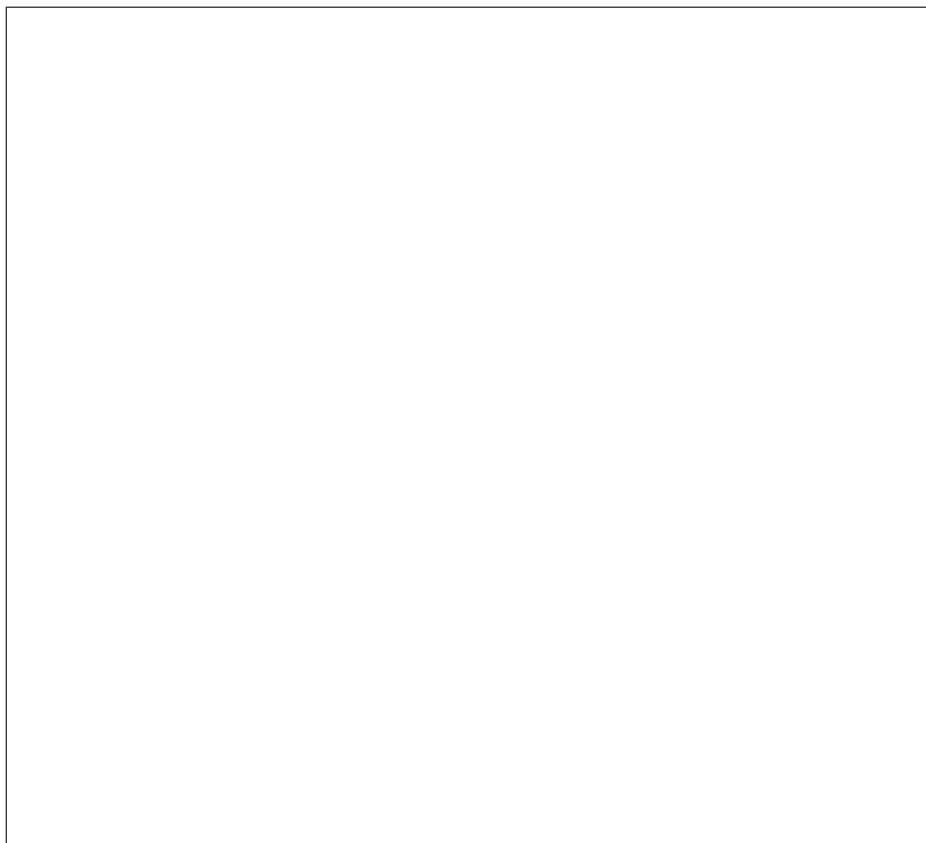
A
de-
ploy
tem-
plate

abstract

Cre-
ate
a
new
node

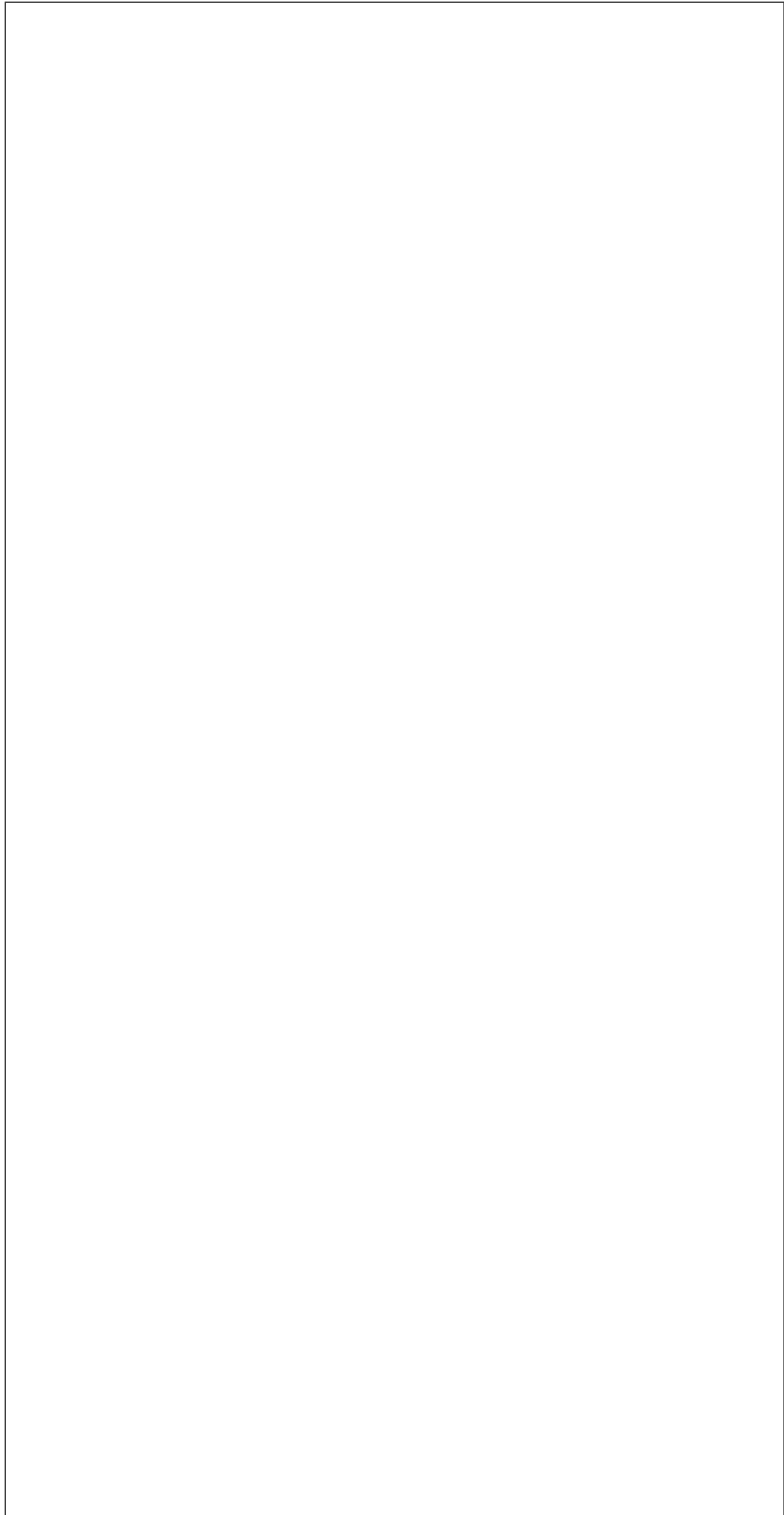
Param
val
A
dict
con-
tain-
ing
sev-
eral
item
used
to
iden-
tify
and
track
the
node
and
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
ues
con-
tains
tags
or
trait

Returns

A
node

abstract

Cre-
ate
a
new
port

Parameter

valu
Dict
of
val-
ues.

abstract

Cre-
ate
a
new
port
grou

Parameter

valu
Dict
of
val-

created_at updated_at

ues
with
the
fol-
low-
ing
keys
id
uuid
nam
node
ad-
dres
ex-
tra

Returns

A
port
grou

Raises

Port
grou
pli-
cate
Nam

Raises

Port
grou
MA
read
ists

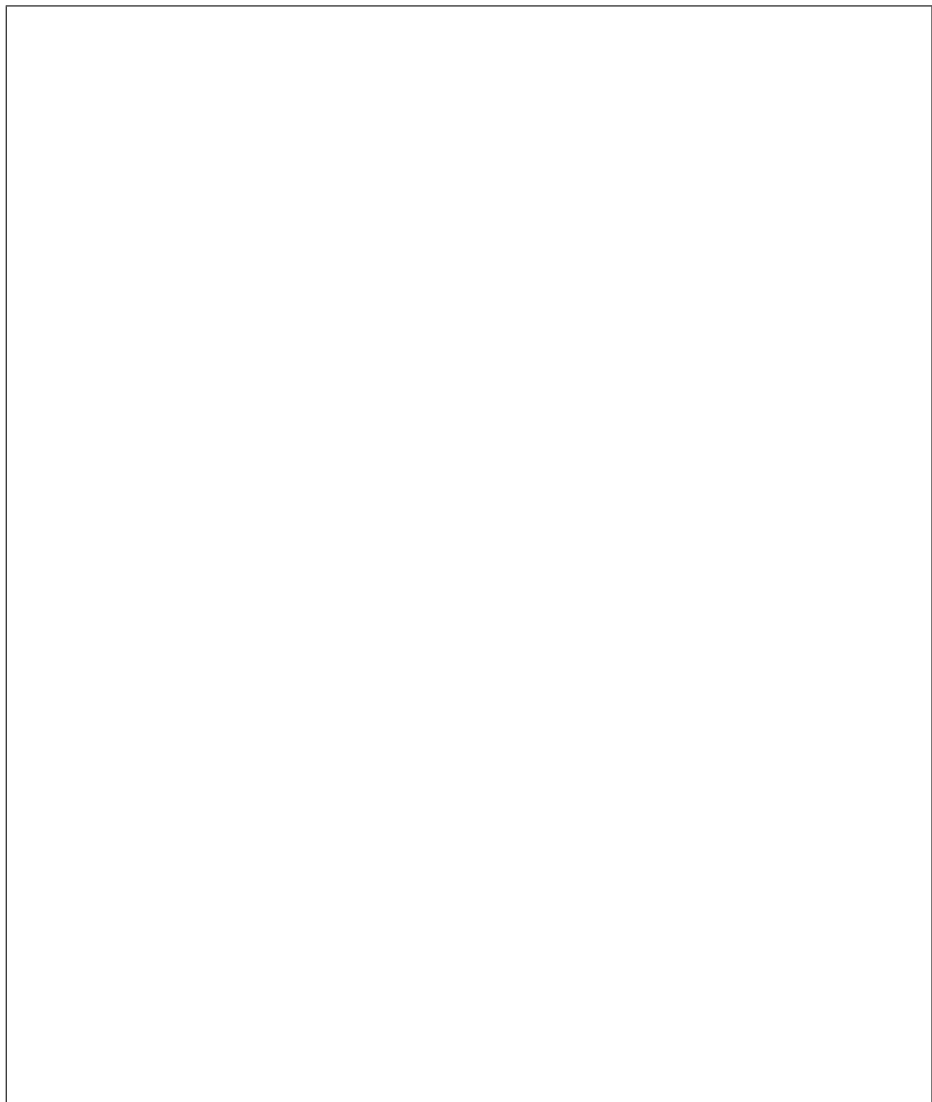
Raises

Port
grou
read
ists

abstract

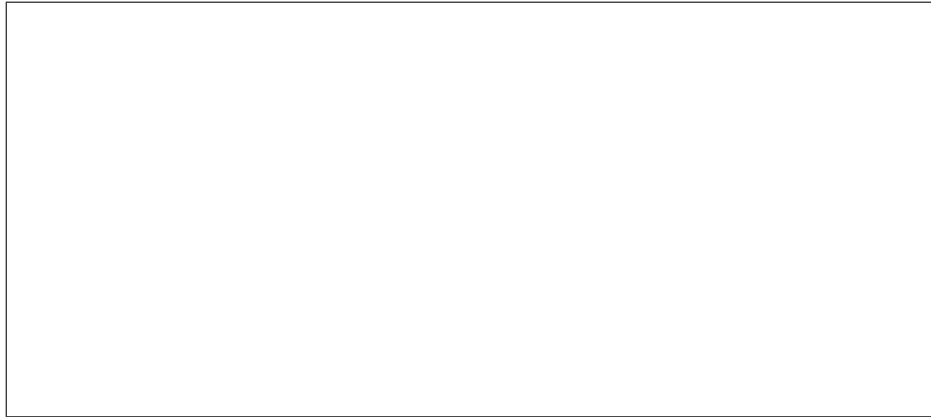
Cre-
ate
a
new
vol-
ume
con-
nec-
tor.

Param
com
Dic-
tio-
nary
con-
tain-
ing
in-
for-
ma-
tion
about
the
con-
nec-
tor.
Ex-
am-
ple:



(continues on next page)

(continued from previous page)



ists with a matching type and connector_id.

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
a
con-
nec-
tor
al-
read
ex-

Raises

Vol-
ume
Con
nec-
torA
read
ists
If

ready exists.

ample:



(continues on next page)

a
vol-
ume
con-
nec-
tor
with
the
sam
UUI
al-

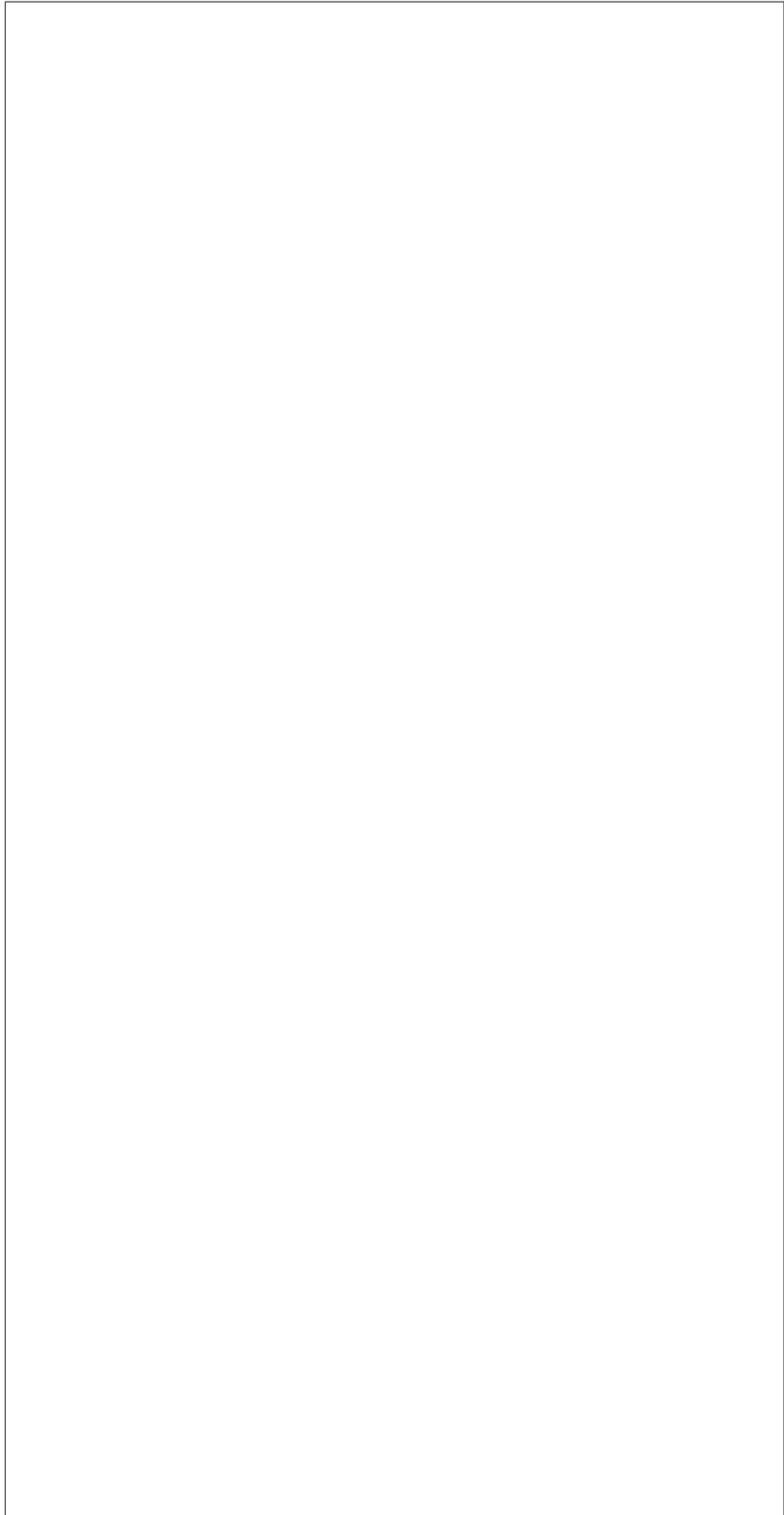
abstract

Cre-
ate
a
new
vol-
ume
tar-
get.

Parame

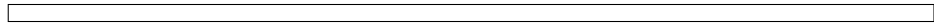
tar
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
the
vol-
ume
tar-
get.
Ex-

(continued from previous page)



(continues on next page)

(continued from previous page)



the same boot index and node ID.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

Raises

Vol-
ume
ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with
the
sam
UUI
ex-
ists.

abstract

Dele
a
list
of
BIO
set-
ting

Parame

- **node**
The
node
id.
- **name**
List
of
BIO
set-
ting
name
to
be
dele

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
any
of
BIO
set-
ting
name

is
not
foun

abstract

Dele
spec
i-
fied
tag
from
the
node

Parame

- **node**
The
id
of
a
node

- **tag**
A
tag
strin

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

Nod
Tag-
Not-
Foun
if
the
tag
is
not
foun

abstract

Dele
spec
i-
fied
trait
from
the
node

Parame

- **node**
The
id
of
a
node

- **tra**
A
trait
strin

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

Nod
Trai
Not-
Foun
if
the
trait
is
not
foun

abstract

De-
stroy

an
al-
lo-
ca-
tion.

Parame

all
Al-
lo-
ca-
tion
ID

Raises

Al-
lo-
ca-
tion-
Not-
Four

abstract

De-
stroy
a
chas
sis.

Parame

cha
The
id
or
the
uuid
of
a
chas
sis.

abstract

De-
stroy
a
de-
ploy
men
tem-
plate

Parame

temp

ID
of
the
de-
ploy
men
tem-
plate
to
de-
stroy

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

abstract

De-
stroy
a
node
and
its
as-
so-
ci-
ated
re-
sour

De-
stroy
a
node
in-
clud
ing
any
as-
so-

nectors, and volume targets.

ci-
ated
port
port
port
grou
tags
trait
vol-
ume
con-

Parame
nod
The
ID
or
UUI
of
a
node

abstract
De-
stroy
an
port

Parame
por
The
id
or
MA
of
a
port

abstract
De-
stroy
a
port
grou

Parame
por
The
UUI
or
MA
of

a
port
grou

Raises

Port
grou
Notl

Raises

Port
grou
Not-
Four

abstract

De-
stroy
a
vol-
ume
con-
nec-
tor.

Parame

iden
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume

does not exist.

con-
nec-
tor
with
the
spec
i-
fied
iden

abstract

De-
stroy
a
vol-
ume
tar-
get.

Parameter

identifier
The
UUID
or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four
if
a
vol-
ume
tar-
get
with
the
spec
i-

exist.

fied
iden
does
not

abstract

Re-
triev
hard
ware
type
for
the
reg-
is-
tere
and
ac-
tive
con-
duc-
tors.

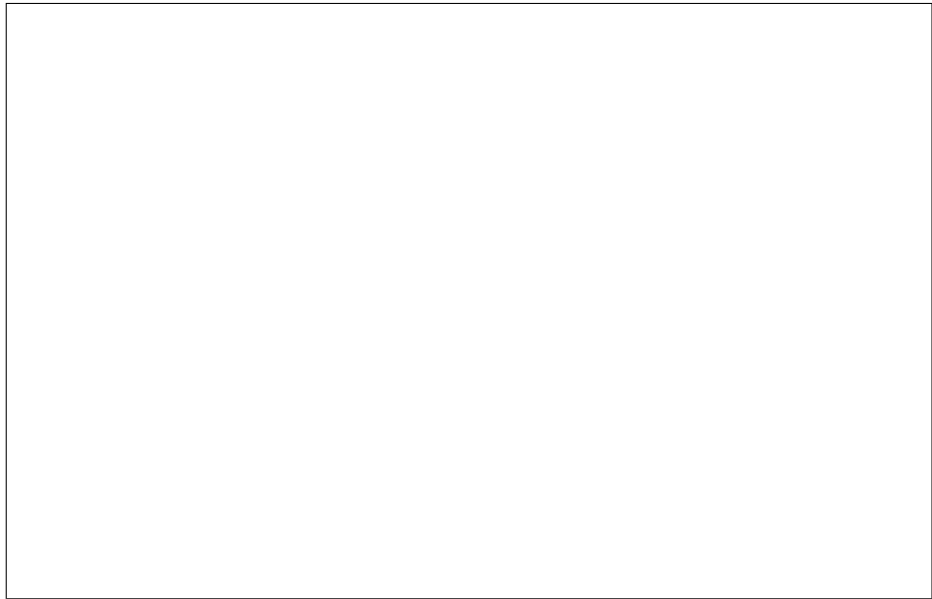
Paramete

use.
Whe
to
fac-
tor
con-
duc-
tor_
into
the
keys

Returns

A
dict
whic
map
hard
ware
type
nam
to
the
set
of

ple:



host
whic
sup-
port
then
For
ex-
am-

abstract

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all
The
id
of
an
al-
lo-
ca-
tion.

Returns

An al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Four

abstract

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

name
The
log-
i-
cal
name
of
an
al-
lo-
ca-
tion.

Returns

An al-
lo-
ca-
tion.

Raises

Al-
lo-

ca-
tion-
Not-
Four

abstract

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all
The
uuid
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion-
Not-
Four

abstract

Re-
turn
a
list
of
al-

lo-
ca-
tion:

Parameter

- **filters**
Filters to apply. Default to Non

node_uuid
uuid of node

state
allocation state

resource_class
request resource class

- **limit**
Maximum number of allocations to return

mar
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
al-
lo-
ca-

tions

abstract

Re-
triev
BIO
set-
ting
valu

Paramete

- **node**
The
node
id.

- **name**
Strin
con-
tain-
ing
nam
of
BIO
set-
ting
to
be
re-
triev

Returns

The
BIO
Set-
ting
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
the
BIO
set-
ting
is
not
foun

abstract

Re-
triev
BIO
set-
tings
of
a
give
node

Paramete

node
The
node
id.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
foun

abstract

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

**Paramete
cha**

The
id
of
a
chas
sis.

Returns

A
chas
sis.

abstract

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

**Paramete
cha**

The
uuid
of
a
chas
sis.

Returns

A
chas
sis.

abstract

Re-
turn
a
list
of
chas
sis.

Parame

- **lim**
Max
i-
mun
num
ber
of
chas
sis
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sor**
At-
tribu
by
whic
re-

sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Re-
triev
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam
of
the
con-
duc-
tor
ser-
vice
- **onl**
Spec

online field is ignored if this value is set to None.

meet the specified online expectation.

ify
the
fil-
ter
valu
on
the
*on-
line*
field
whe
quer
ing
con-
duc-
tors.
The

Returns

A
con-
duc-
tor.

Raises

Con-
duc-
torN
Four
if
the
con-
duc-
tor
with
give
host
nam
does
not
ex-
ist
or
does

abstract

Re-
turn
a
list

of
con-
duc-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
con-
duc-
tors
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be

sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Re-
triev
a
de-
ploy
men
tem-
plate
by
ID.

Paramete

temp
ID
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy

tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

abstract

Re-
triev
a
de-
ploy
men
tem-
plate
by
nam

Paramete

tem
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does

not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

abstract

Re-
triev
a
de-
ploy
men
tem-
plate
by
UUI

Paramet

tem
UUI
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy-
tem-
plate

abstract

Re-
triev
a
list
of
de-
ploy
men
tem-
plate

Parame

- **lim**
Max
i-
mun
num
ber
of
de-
ploy
tem-
plate
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
de-
ploy
tem-
plate

abstract

Re-
turn
a
list
of
de-
ploy
men
tem-

plate
with
one
of
a
list
of
nam

Parame

nam
List
of
nam
to
fil-
ter
by.

Returns

A
list
of
de-
ploy
tem-
plate

abstract

Re-
turn
a
node

Parame

nod
The
id
of
a
node

Returns

A
node

abstract

Re-
turn
a
node

Parame

ins
The
in-
stan-
uuid
to
sear-
for.

Returns
A
node

Raises
In-
stan-
ceNo-
Four
if
the
in-
stan-
is
not
foun-

Raises
In-
valic
UID
if
the
in-
stan-
uuid
is
in-
valic

abstract
Re-
turn
a
node

Parame
node
The
log-
i-
cal
nam
of

a
node

Returns

A
node

abstract

Find
a
node
by
any
match
ing
port
ad-
dres

Parameter

add:
list
of
port
ad-
dres
(e.g.
MA

Returns

Nod
ob-
ject.

Raises

Nod
Not-
Four
if
none
or
sev-
eral
node
are
foun

abstract

Re-
turn
a
node

Parameter
node
The
uuid
of
a
node

Returns
A
node

abstract
Re-
turn
a
list
of
node

Parameter

- **fil-**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associ
True
|
Fals

reserv
True
|
Fals

maint
True
|
Fals

chassis
uuid
of

chas
sis

driver
driv
nam

provis
pro-
vi-
sion
state
of
node

provis
node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in
sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-

ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Get
node
tags
base
on
its
id.

Paramete

node
The
id
of

a
node

Returns

A
list
of
Nod
Tag
ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
foun

abstract

Get
node
trait
base
on
its
id.

Parame

node
The
id
of
a
node

Returns

A
list
of
Nod
Trai
ob-
jects

Raises

Nod
Not-
Four

if
the
node
is
not
foun

abstract

Get
spe-
cific
colu
for
mat
ing
node

Re-
turn
a
list
of
the
spec
i-
fied
colu
for
all
node
that
mat
the
spec
i-
fied
fil-

ters.

Parame

- **col:**
List
of
col-
umn
nam
to
re-
turn

De-
fault
to
id
col-
umn
whe
colu
==
Non

- **fil:**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associ:
True
|
Fals

reserv:
True
|
Fals

reserv:
[com
duc-
tor1
con-
duc-
tor2

maint:
True
|
Fals

retired:
True
|
Fals

chassis:
uuid
of

chas
sis

driver
driv
nam

provis
pro-
vi-
sion
state
of
node

provis
node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in
sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-

ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
tu-
ples
of
the
spec
i-
fied
colu

abstract

Re-
turn
ob-
jects
with
ver-
sion
that
are
not
the
spec-
i-
fied
ver-
sion

Parame

- **mod**
the
nam
of
the
mod
(clas
of
de-
sired
ob-
jects
- **ver:**
list
of
ver-
sion
of
ob-
jects
not
to
be
re-
turn

Returns

list
of

the
DB
ob-
jects

Raises

Iron
icEx
cep-
tion
if
there
is
no
class
as-
so-
ci-
ated
with
the
nam

abstract

Get
a
list
con-
duc-
tors
that
are
of-
fine
(dea

Parame

file
A
field
to
re-
turn
host
nam
by
de-
fault

Returns

A
list

of
re-
ques
field
of
of-
flin
con-
duc-
tors.

abstract

Get
a
list
con-
duc-
tor
host
nam
that
are
on-
line
and
ac-
tive.

Returns

A
list
of
con-
duc-
tor
host
nam

abstract

Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

add:
The
MA
ad-
dres
of
a
port

Returns
A
port

abstract
Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame
por
The
id
of
a
port

Returns
A
port

abstract
Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame
por

The
nam
of
a
port

Returns

A
port

abstract

Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Paramete

por

The
uuid
of
a
port

Returns

A
port

abstract

Re-
turn
a
list
of
port

Paramete

- **lim**
Max
i-
mun
num
ber

of
port
to
re-
turn

- **mark**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Re-
turn
a
net-
work
port
group
rep-
re-
sen-
ta-
tion.

Paramete

- **add**
The
MA
ad-
dres
of
a
port
grou

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns

A
port
grou

Raises

Port
grou
Not-
Four

abstract

Re-

turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Four

abstract

Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

nam
The
log-
i-
cal
nam

of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Four

abstract

Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
uuid
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Four

abstract

Re-
turn

a
list
of
port
group

Parame

- **lim**
Max
i-
mun
num
ber
of
port
group
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be

sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns

A
list
of
port
grou

abstract

List
all
the
port
grou
for
a
give
node

Parame

- **nod**
The

in-
te-
ger
node
ID.

- **lim**
Max
i-
mun
num
ber
of
port
grou
to
re-
turn

- **marl**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

-

sort
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns
A
list
of
port
grou

abstract

List
all
the
port
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node

ID.

- **limit**
Maximum number of ports to return

- **mark**
The last item of the previous page we return the next result set.

- **sort**
Attribute by which results should be sorted

- **sort**
direction in

whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
port

abstract

List
all
the
port
for
a
give
port
grou

Parame

- **por-**
The
in-
te-
ger
port
grou
ID.
- **lim**
Max
i-
mun
num
ber
of
port
to
re-
turn
-

mar

The last item of the previous page we return the next result set.

•

sort

Attribute by which results should be sorted

•

sort

Direction in which results should be sorted (ascending or descending)

Returns

A list of port

abstract

Re-
turn
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Param

db_
The
in-
te-
ger
data
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID.

Raises

Vol-
ume
Con
nec-
torN
Four
If

is not found.

a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID

abstract

Re-
turn
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

con
The
UI
of
a
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UI

is not found.

Raises

Vol-
ume
Con-
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UU

abstract

Re-
turn
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-

turn

- **mark**
The last item of the previous page we return the next result set.

- **sort**
Attribute by which results should be sorted

- **sort**
Direction in which results should be sorted (ascending/descending)

- **prop**
The as-

so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
If
sort
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
con-
nec-
tors

for
a
give
node

Parame

- **node**
The
in-
te-
ger
node
ID.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of
Vol
ob-
jects

Returns
A
list
of
vol-

ume
con-
nec-
tors.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
If
sort
does
not
ex-
ist.

abstract

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parameter

db_:
The
data
pri-
mar-
key
(in-
te-
ger)
ID
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists.

abstract

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Paramete

uui
The
UUI
of
a
vol-
ume
tar-
get.

Returns

A

vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
UI
ex-
ists.

abstract

Re-
turn
a
list
of
vol-
ume
tar-
gets.

Paramete

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn

- **mark**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte
- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc
- **pro**
The
as-
so-
ci-

ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
tar-
gets
for
a
give
node

Parame

- **node**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sor**
At-
tribu
by

whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-

valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
tar-
gets
for
a
give
vol-
ume
id.

Paramete

- **vol**
The
UI
of
the
vol-
ume
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-

gets
to
re-
turn

- **mark**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
tar-
gets.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

abstract

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor.

Returns

List
of

Con
ob-
jects

abstract

List
reg-
is-
tere
hard
ware
in-
ter-
face
for
give
hard
ware
type

This
is
re-
stric
to
only
ac-
tive
con-
duc-
tors.
:par
hard
ware
list
of
hard
ware
type
to

filter by. :returns: list of `ConductorHardwareInterfaces` objects.

abstract

Che
if
the
spec
i-
fied
tag
ex-

ist
on
the
node

Parameters

- **node**
The
id
of
a
node

- **tag**
A
tag
string

Returns

True
if
the
tag
ex-
ists
oth-
er-
wise
False

Raises

Node
Not-
Found
if
the
node
is
not
found

abstract

Check
if
the
spec
i-
fied
trait

ex-
ists
on
the
node

Parame

- **node**
The
id
of
a
node
- **tra**
A
trait
strin

Returns

True
if
the
trait
ex-
ists
oth-
er-
wise
Fals

Raises

Nod
Not-
Four
if
the
node
is
not
foun

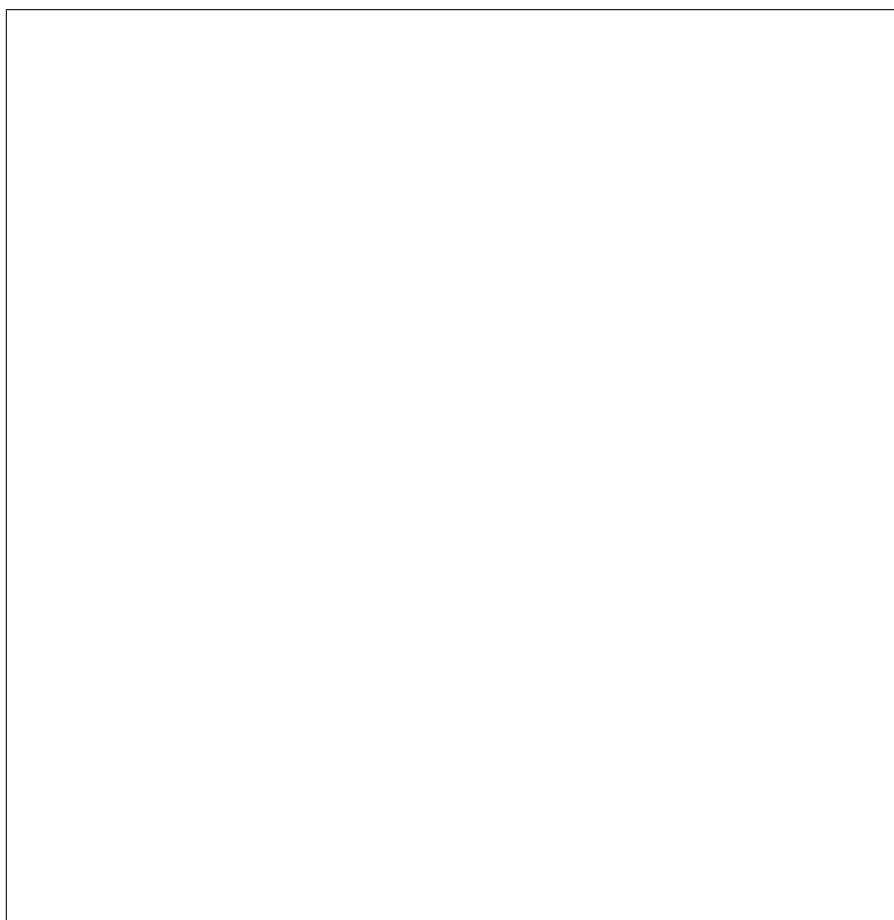
abstract

Reg
is-
ter
an
ac-
tive

con-
duc-
tor
with
the
clus
ter.

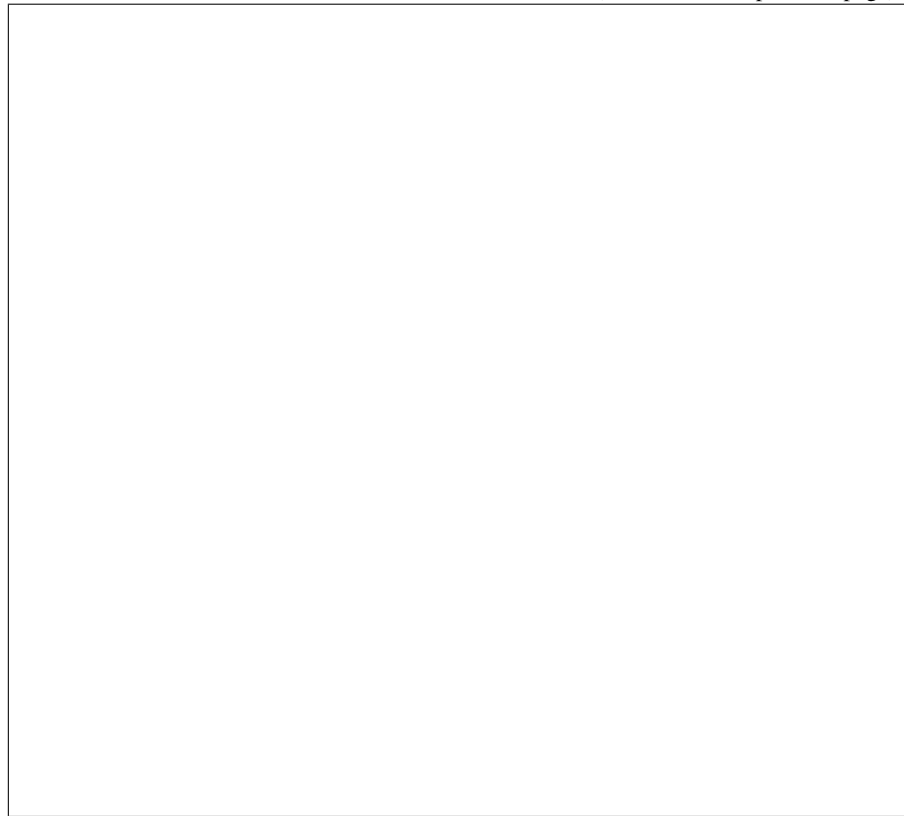
Parame

- **val**
A
dict
of
val-
ues
whic
mus
con-
tain
the
fol-
low-
ing:



(continues on next page)

(continued from previous page)



line record is found. When true, will overwrite the existing record. Default: False.

- **upd.**
When
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion
when
a
con-
flict-
ing
on-

Returns

A
con-
duc-
tor.

Raises

Con
duc-
torA
read
is-
tere

abstract

Reg
is-
ters
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

- **con**
Data
ID
of
con-
duc-
tor
to
reg-
is-
ter
for.

- **har**
Nam
of
hard
ware
type
for
the
in-
ter-

face

- **int**
Type
of
in-
ter-
face
e.g.
de-
ploy
or
boot

- **int**
List
of
in-
ter-
face
nam
to
reg-
is-
ter.

- **def**
Strin
the
de-
fault
in-
ter-
face
for
this
hard
ware
type
and
in-
ter-
face
type

Raises
Con
duc-
torH
ware

in the combination of all parameters is already registered.

ter-
face
sAl-
read
is-
tere
if
at
leas
one
of
the
in-
ter-
face

abstract

Re-
lease
the
rese
va-
tion
on
a
node

Paramet

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **node**
A
node
id
or

uuid

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
re-
serv
by
an-
othe
host

Raises

Nod
Not-
Lock
if
the
node
was
foun
to
not
have
a
rese
va-
tion
at
all.

abstract

Re-
serv
a
node
To

performed, mark it reserved by this host.

pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the
give
Nod
whil
a
Task
is

Parame

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**
A
node
id
or
uuid

Returns

A
Nod
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
al-
read
re-
serv

abstract

Re-
plac
all
of
the
node
tags
with
spec
i-
fied
list
of
tags
This
ig-
nore
du-
pli-
cate
tags
in
the
spec
i-
fied
list.

Parameters

- **node**
The id of a node
- **tag**
List of tags

Returns

A list of NodeTag objects

Raises

NodeNotFoundError if the node is not found

abstract

Replace all of the node traits with specified list of traits

This
ig-
nore
du-
pli-
cate
trait
in
the
spec
i-
fied
list.

Parameters

- **node**
The
id
of
a
node
- **traits**
List
of
traits
- **version**
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
Node
Trait
ob-
jects

Raises

In-
valid
Pa-

limit.

ram-
e-
ter-
Valu
if
set-
ting
the
trait
wou
ex-
ceed
the
per-
node
trait

Raises

Nod
Not-
Four
if
the
node
is
not
foun

abstract

Do
a
take
over
for
an
al-
lo-
ca-
tion.
The
al-
lo-
ca-
tion
is
only
up-
date
if
the

thus guarding against races.

cation.

old
con-
duc-
tor
mat
the
pro-
vide
valu

Paramete

- **all**
Al-
lo-
ca-
tion
ID
- **old**
The
con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
rent
con
of
the
al-
lo-
- **new**
The
con-
duc-
tor
ID
of
the

new
con

Returns

True
if
the
take
over
was
suc-
cess-
ful,
Fals
oth-
er-
wise

Raises

Al-
lo-
ca-
tion-
Not-
Fou

abstract

Mar
a
con-
duc-
tor
as
ac-
tive
by
up-
dat-
ing
its
up-
date
prop
erty.

Paramete

hos
The
host
nam
of
this

con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

abstract

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning
by
up-
dat-
ing
its
pro-
vi-
sion
prop
erty.

Parameter

node
The
id
of
a
node

Raises

Nod
Not-
Four

abstract

Re-
mov
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry
im-
me-
di-
ately

Parame

hos

The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

abstract

Un-
reg-
is-
ters
all
hard
ware
in-
ter-

face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor
to
un-
reg-
is-
ter
for.

abstract

Re-
mov
all
tags
of
the
node

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

abstract

Re-

mov
all
trait
of
the
node

Parame

nod

The
id
of
a
node

Raises

Nod
Not-
Four
if
the
node
is
not
foun

abstract

Up-
date
prop
er-
ties
of
an
al-
lo-
ca-
tion.

Parame

- **all**
Al-
lo-
ca-
tion
ID

- **val**
Dict

allocation

of
val-
ues
to
up-
date

- **upd**
If
True
and
node
is
up-
date
up-
date
the
node
with
in-
stan-
and
trait
from
the

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-
tion.
Not-
Four

Raises

Al-
lo-
ca-
tion)
pli-
cate
Nam

Raises

In-
stan-
As-
so-
ci-
ated

Raises

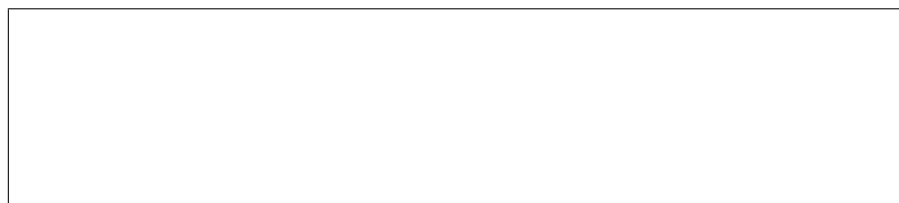
Nod
As-
so-
ci-
ated

abstract

Up-
date
a
list
of
BIO
Set-
ting
reco

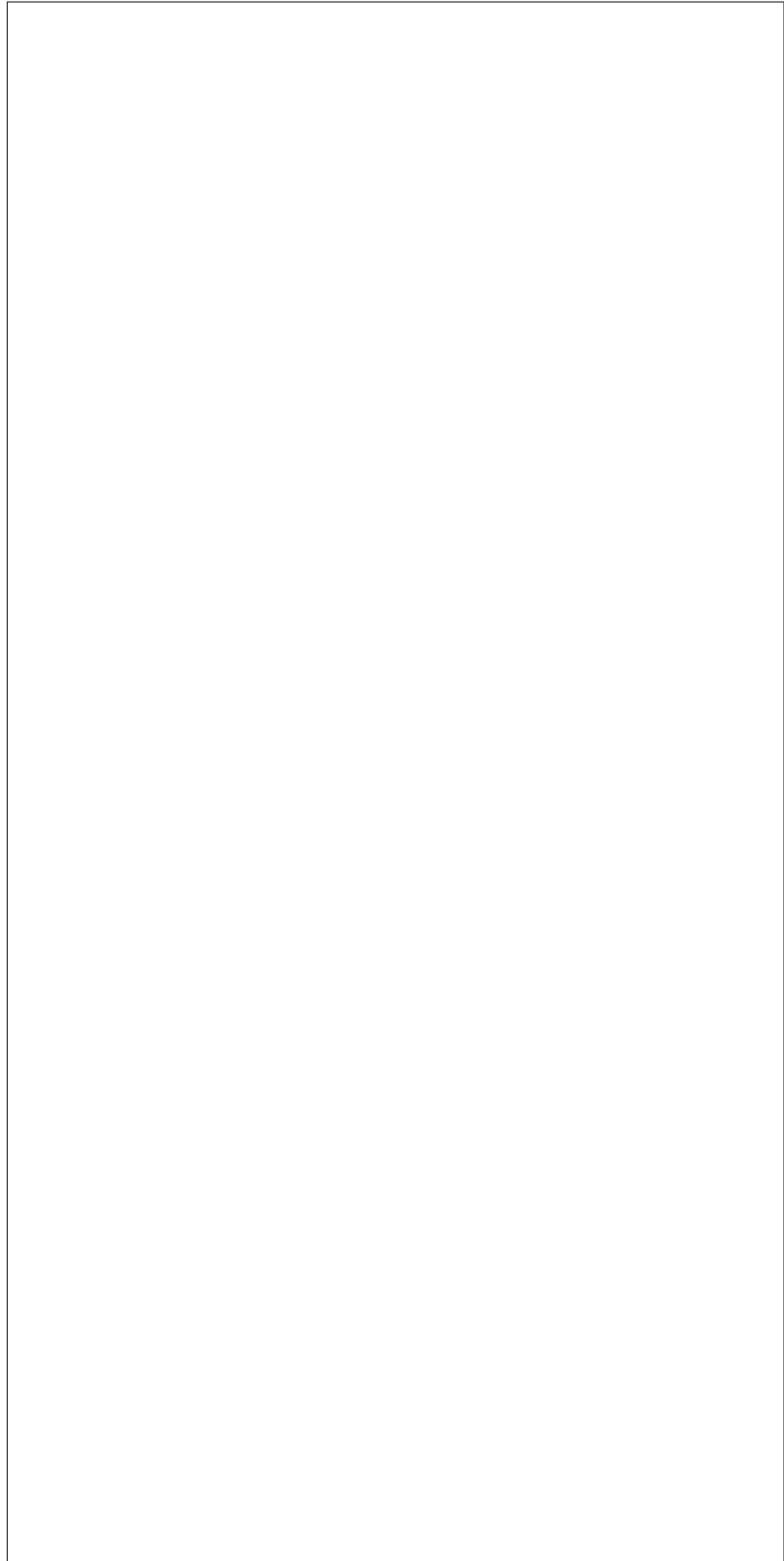
Parame

- **nod**
The
node
id.
- **set**
A
list
of
BIO
Set-
tings
to
be
up-
date



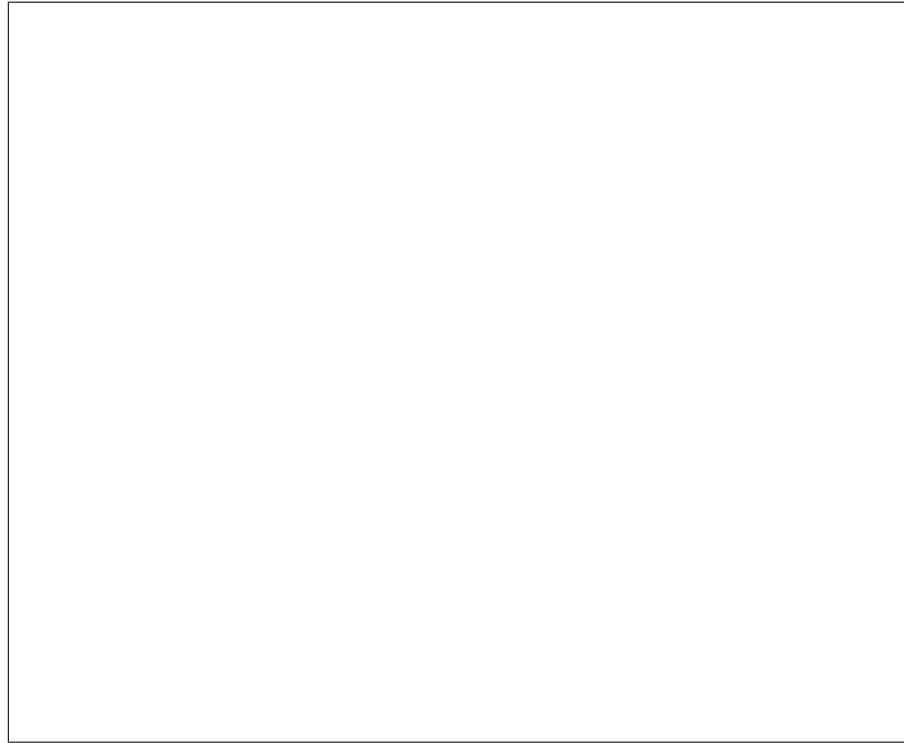
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



•

ver:
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
any
of
the
set-
ting-
is
not
foun

abstract

Up-
date
prop
er-
ties
of
an
chas
sis.

Parame

-

cha
The
id
or
the
uuid
of
a
chas
sis.

-

valu
Dict
of
val-
ues
to
up-
date

Returns

A
chas
sis.

abstract

Up-
date
a
de-
ploy
men
tem-
plate

Parame

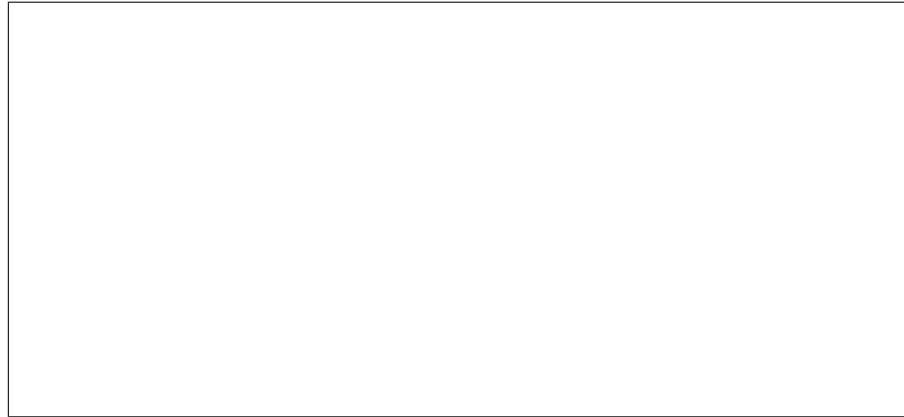
- **temp**
ID
of
the
de-
ploy
men
tem-
plate
to
up-
date

- **val**
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



(continues on next page)

(continued from previous page)



Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A

de-
ploy
tem-
plate

abstract

Up-
date
prop
er-
ties
of
a
node

Paramete

- **node**
The
id
or
uuid
of
a
node

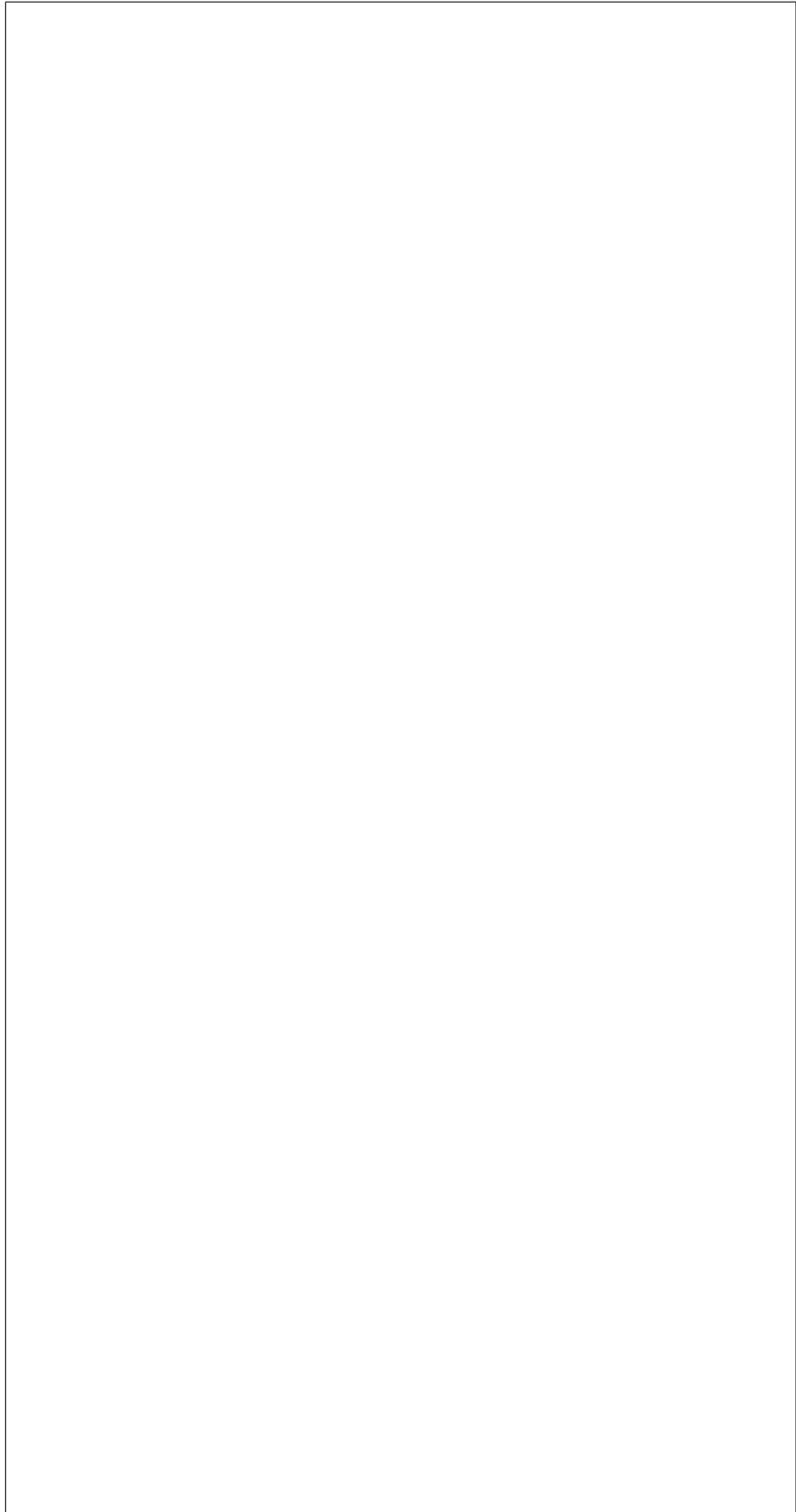
- **value**
Dict
of
val-
ues
to
up-
date
May
be
a
par-
tial
list,
eg.
when
set-
ting
the

properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns
A

node

Raises

Nod
As-
so-
ci-
ated

Raises

Nod
Not-
Four

abstract

Up-
date
prop
er-
ties
of
an
port

Parame

-

por
The
id
or
MA
of
a
port

-

valu
Dict
of
val-
ues
to
up-
date

Returns

A
port

abstract

Up-
date

prop-
er-
ties
of
a
port
group

Parameters

- **port**
The
UUID
or
MAC
of
a
port
group
- **value**
Dict
of
val-
ues
to
up-
date
May
con-
tain
the
fol-
low-
ing
keys
uuid
name
node

address extra created_at updated_at

Returns

A
port
group

Raises

In-
valid
Pa-
ram-

e-
ter-
Valu

Raises

Port
grou
Not-
Four

Raises

Port
grou
pli-
cate
Nam

Raises

Port
grou
MA
read
ists

abstract

Up-
date
ob-
jects
to
their
lat-
est
know
ver-
sion

This
scan
all
the
ta-
bles
and
for
ob-
jects
that
are
not
in
their
lat-

dates them to that version.

all the objects will be migrated.

est
ver-
sion
up-

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate
Mus
be
>=
0.
If
zero

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that

the beginning of this call) and 2. the number of migrated objects.

need
to
be
mi-
grate
(at

abstract

Up-
date
prop
er-
ties
of
a
vol-
ume
con-
nec-
tor.

Parame

- **id**
The
UI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-
tor.
- **con**
Dic-
tio-
nary
con-
tain-
ing
the
in-

for-
ma-
tion
about
con-
nec-
tor
to
up-
date

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con-
nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
an-
othe
con-
nec-
tor
al-
read

exists with a matching type and connector_id field.

Raises

Vol-
ume
Con-
nec-
torN
Four
If
a
vol-
ume
con-

does not exist.

nec-
tor
with
the
spec
i-
fied
iden

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
Whe
a
UUI
is
in-
clud
in
con-
nec-
tor_

abstract

Up-
date
in-
for-
ma-
tion
for
a
vol-
ume
tar-
get.

Parame

- **iden**
The
UUI
or
in-

date.

te-
ger
ID
of
a
vol-
ume
tar-
get.

- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
vol-
ume
tar-
get
to
up-

Returns

A
vol-
ume
tar-
get.

Raises

In-
valid
Pa-
ram-
e-
ter
Valu
if
a
UUI
is
in-
clud

the same boot index and node ID.

in
tar-
get_
Raises
Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

Raises
Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
iden
ex-
ists.

ironic.d

Re-
turn
a
DB
API
in-
stan

ironic.db.migration module

Data
setu
and
mi-
gra-
tion
com
man

ironic.d

ironic.d

ironic.d

ironic.d

ironic.d

Mi-
grate
the
data
to
*ver-
sion*
or
the
mos
re-
cent
ver-
sion

ironic.d

Module contents

ironic.dhcp package

Submodules

ironic.dhcp.base module

Ab-
strac

base
class
for
dhcp
prov

class ir

Base
obj

Base
class
for
DHCP
prov
API

clean_dh

Clea
up
the
DHCP
BOC
op-
tions
for
all
port
in
task

Paramete

task
A
Task
ager
in-
stan

Raises

Fail
To-
Cle-
anD
HCF

get_ip_a

Get
IP
ad-
dres

for
all
port
in
task

Parame

task
A
Task
ager
in-
stan

Returns

List
of
IP
ad-
dres
as-
so-
ci-
ated
with
task
port
and
port
grou

abstract

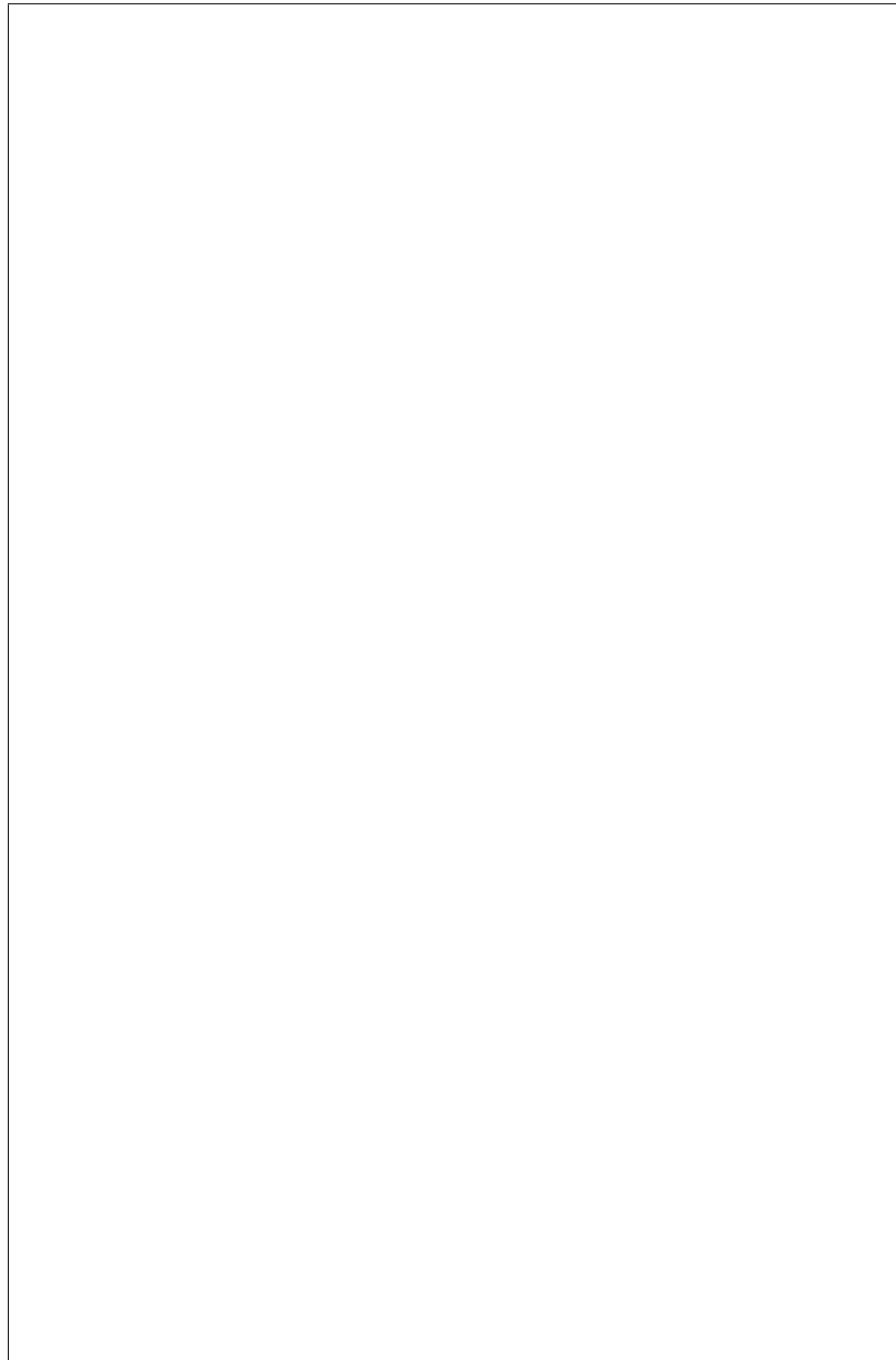
Send
or
up-
date
the
DHCP
BOC
op-
tions
for
this
node

Parame

- **task**
A
Task
ager

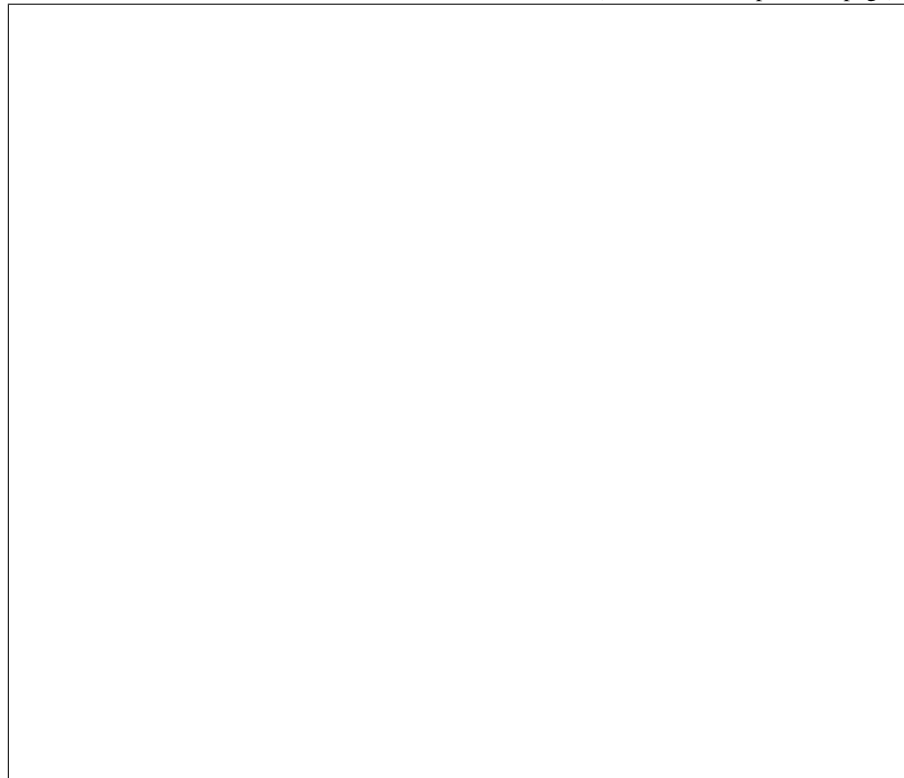
in-
stan

- **opt.**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



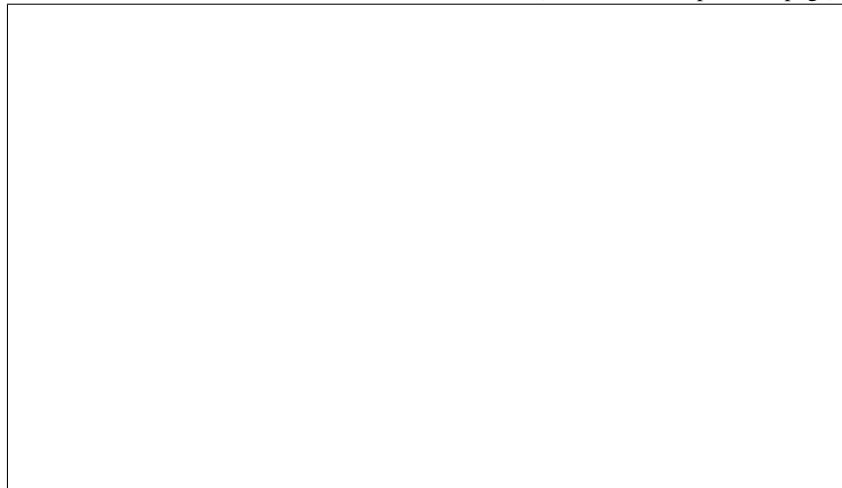
• **vif:**
A
dict
with
keys
port
and
port
group
and
dicts
as
val-
ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port
ob-
jects

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstract

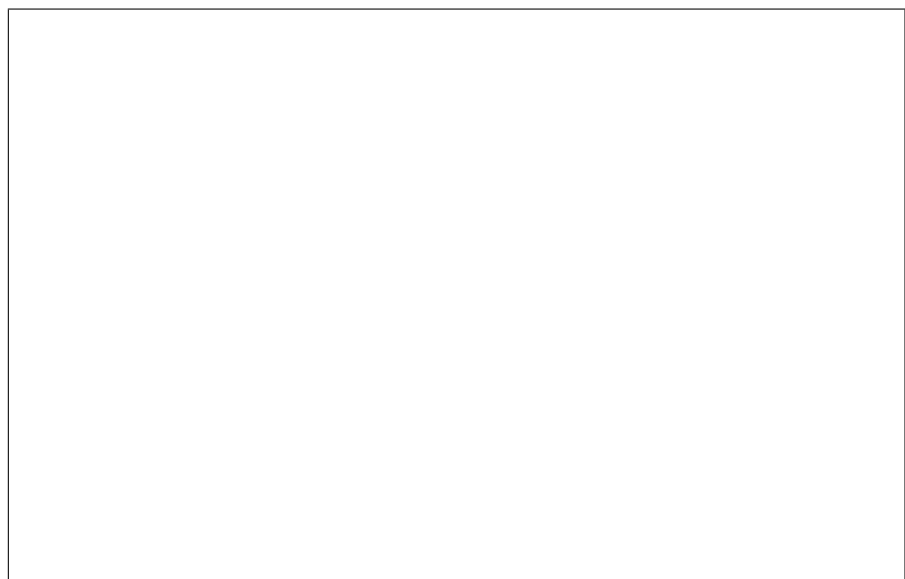
Up-
date
one
or
mor
DH
op-
tion

on
the
spec
i-
fied
port

Parame

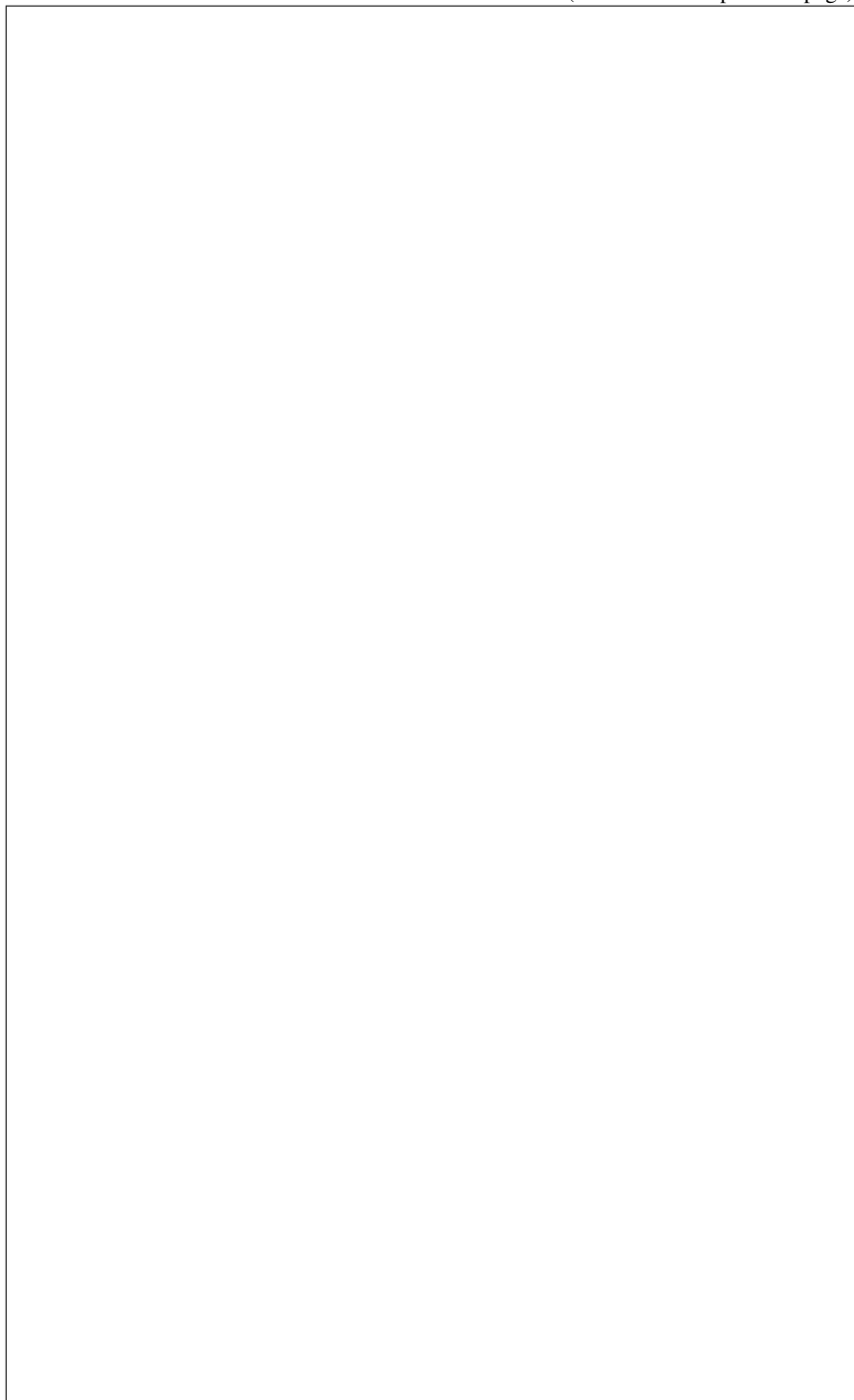
- **por-**
des-
ig-
nate
whic
port
thes
at-
tribu
will
be
ap-
plic
to.

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



- **tok**
An
op-
tion:
au-
then-
ti-
ca-

tion
to-
ken.
Dep
re-
cate
use
con-
text

- **con**
(irc
com
con
Req
re-
ques
con-
text

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

ironic.dhcp.neutron module

class ir

Base
iro
dhc
bas
Bas

API
for
com
mu-
ni-
cat-
ing
to
neu-
tron
2.x
API

get_ip_a

Get
IP
ad-
dres
for
all
port
in
task

Parame

tas
a
Task
ager
in-
stan

Returns

List
of
IP
ad-
dres
as-
so-
ci-
ated
with
task
port

update_c

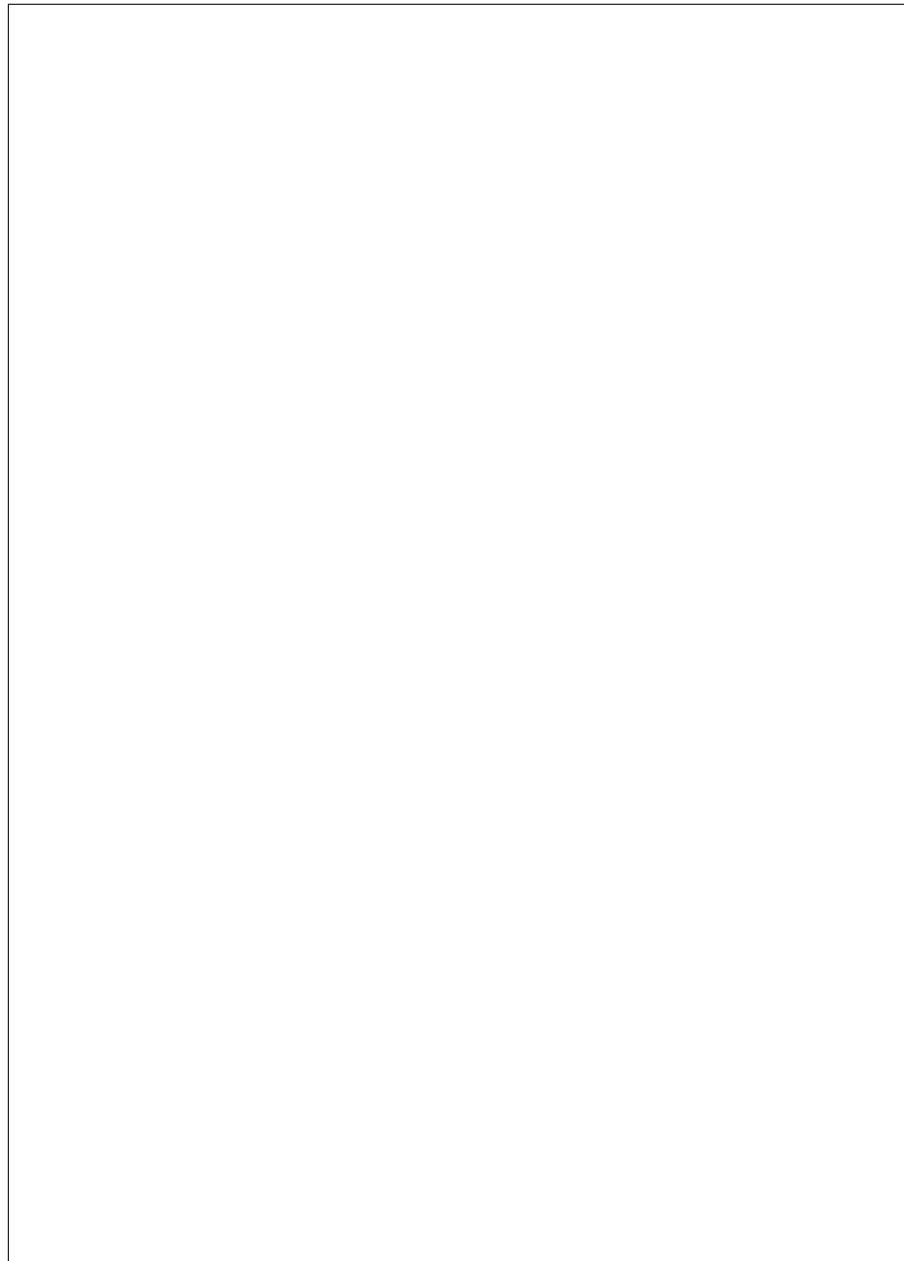
Send
or
up-
date
the
DHCP
BOOTP
op-
tions
for
this
node

Parame

- **tas**

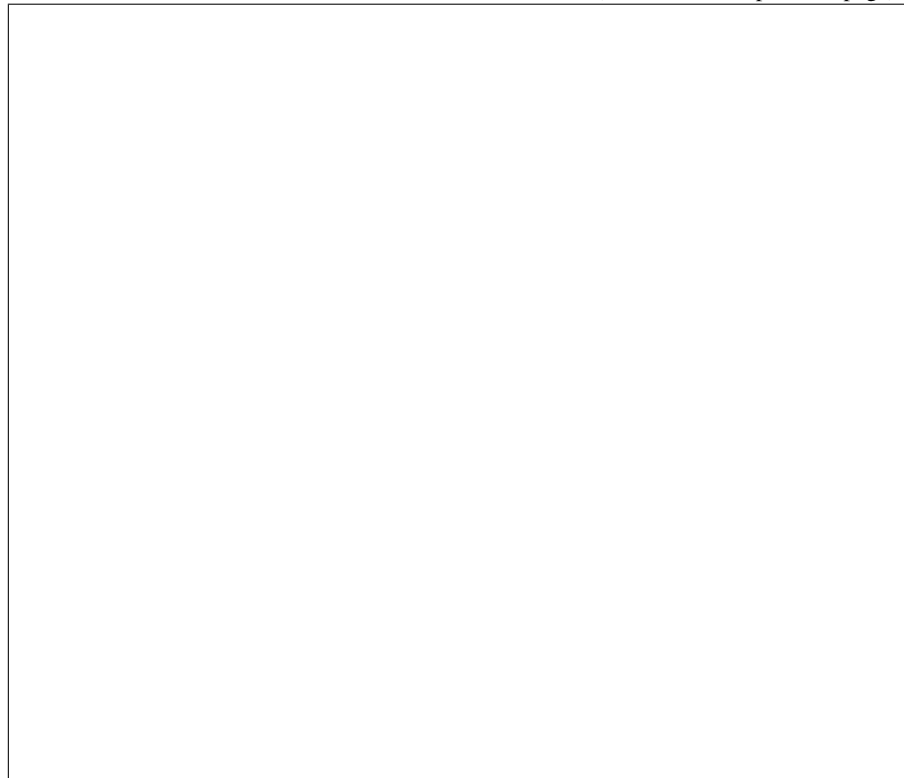
A
Task
ager
in-
stan

- **opt.**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



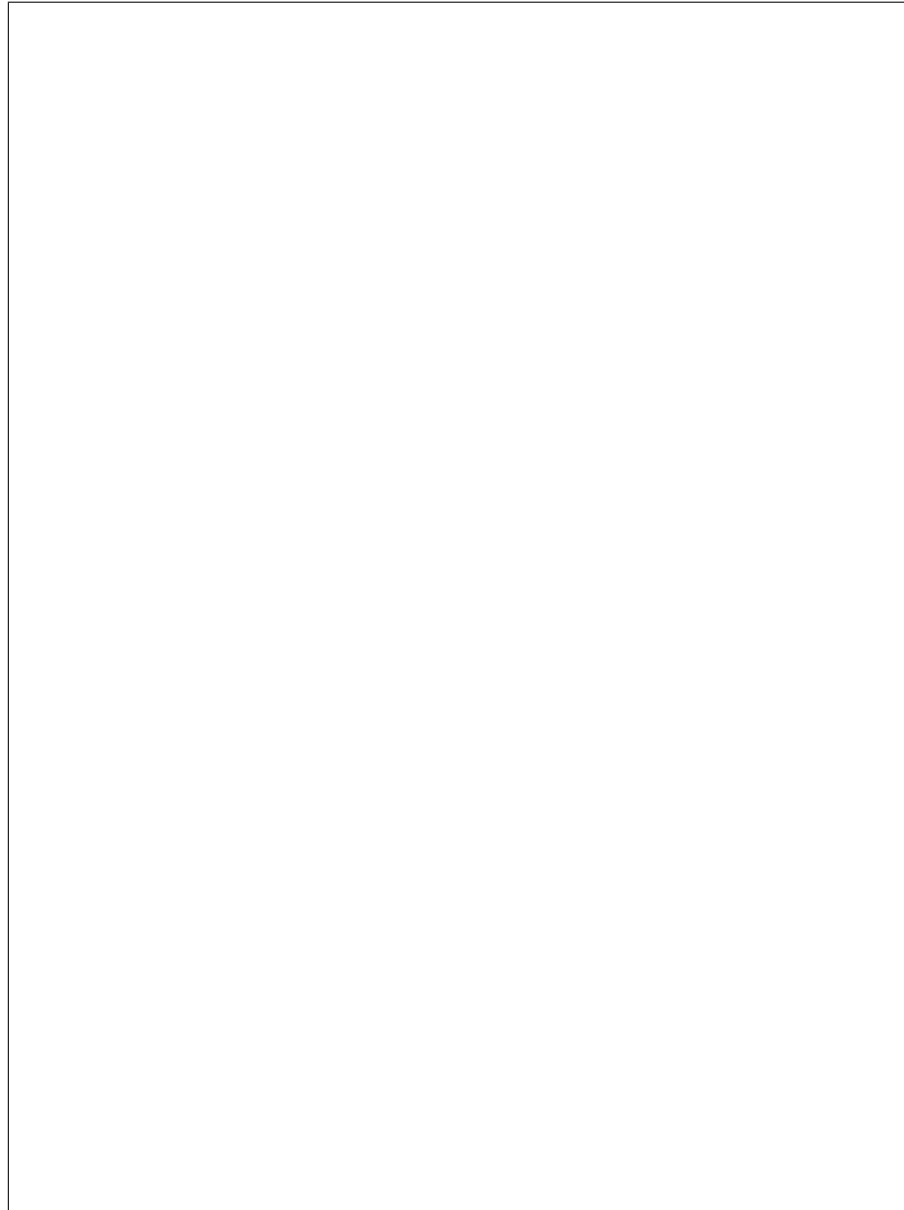
• **vif:**
a
dict
of
Neu
tron
port
dicts
to
up-
date
DHCP
op-
tions
on.
The
port
dict
key

should be Ironic port UUIDs, and the values should be Neutron port UUIDs, e.g.



(continues on next page)

(continued from previous page)



update_p

Up-
date
a
port
at-
tribu

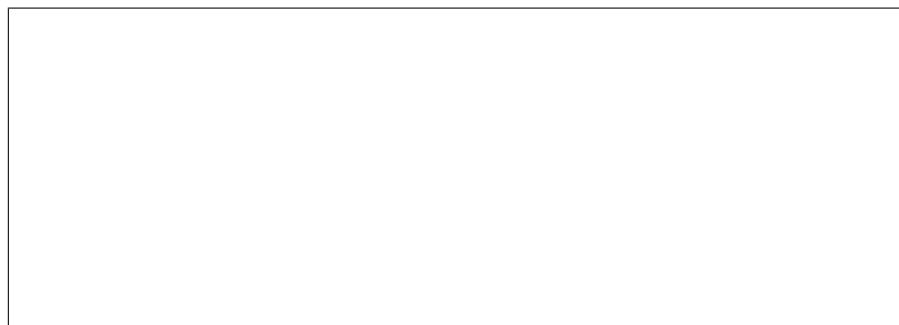
Up-
date
one
or
more
DHCP
op-
tions

on
the
spec
i-
fied
port
For
the
rel-
e-
vant
API

spec, see <https://docs.openstack.org/api-ref/network/v2/index.html#update-port>

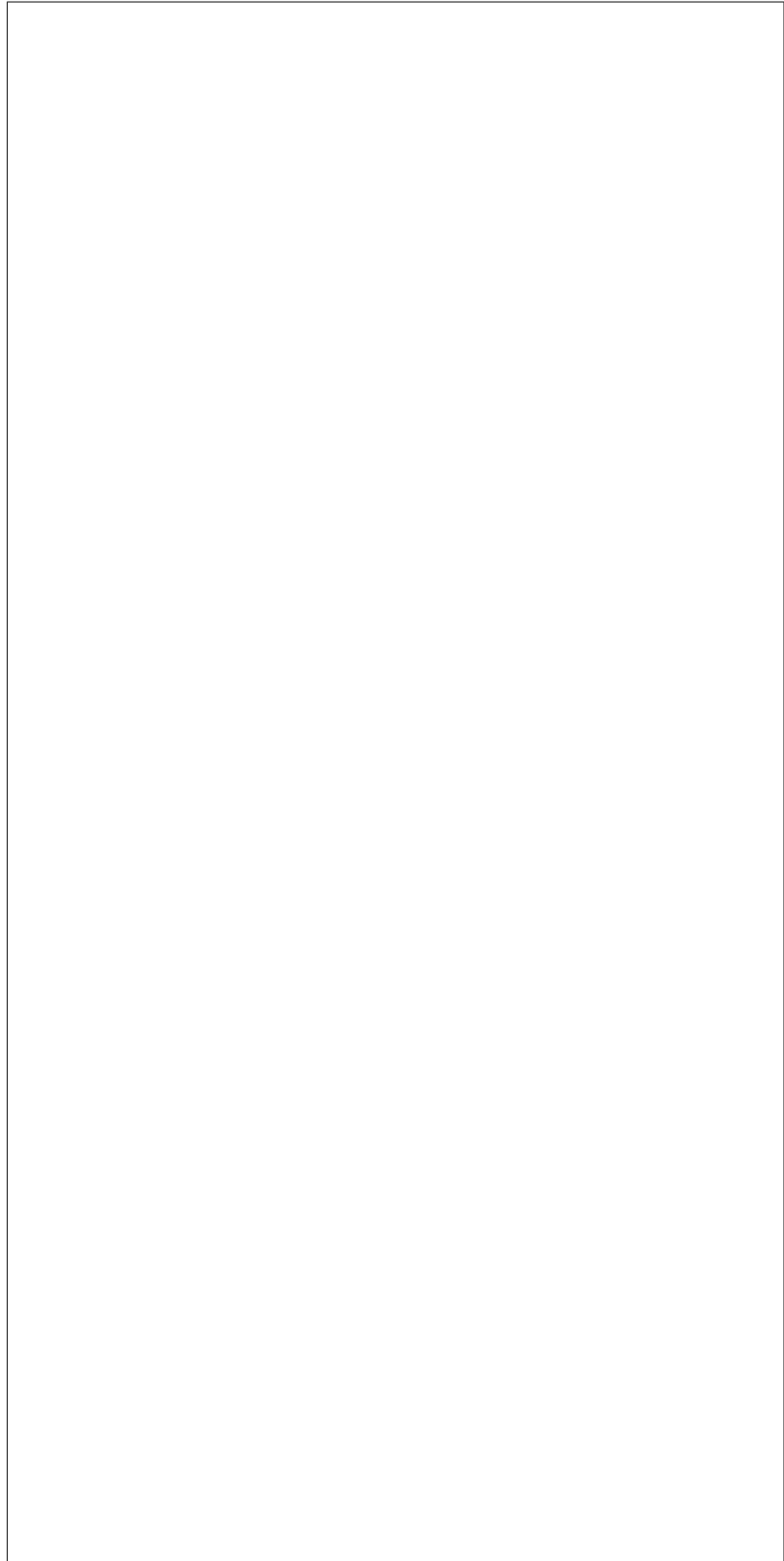
Parameter

- **port**
des-
ig-
nate
whic
port
thes
at-
tribu
will
be
ap-
plie
to.
- **dhcp**
this
will
be
a
list
of
dicts
e.g.



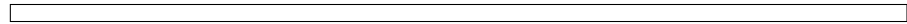
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



- **tok**
op-
tiona
auth
to-
ken.
Dep
re-
cate
use
con-
text.

- **con**
(irc
com
con
Req
re-
ques
con-
text

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

ironic.dhcp.none module

class ir

Base
iro
dhc
bas
Bas
No-
op
DHC
API

get_ip_a

Get
IP
ad-
dres
for
all
port
in
task

Parame

task
A
Task
ager
in-
stan

Returns

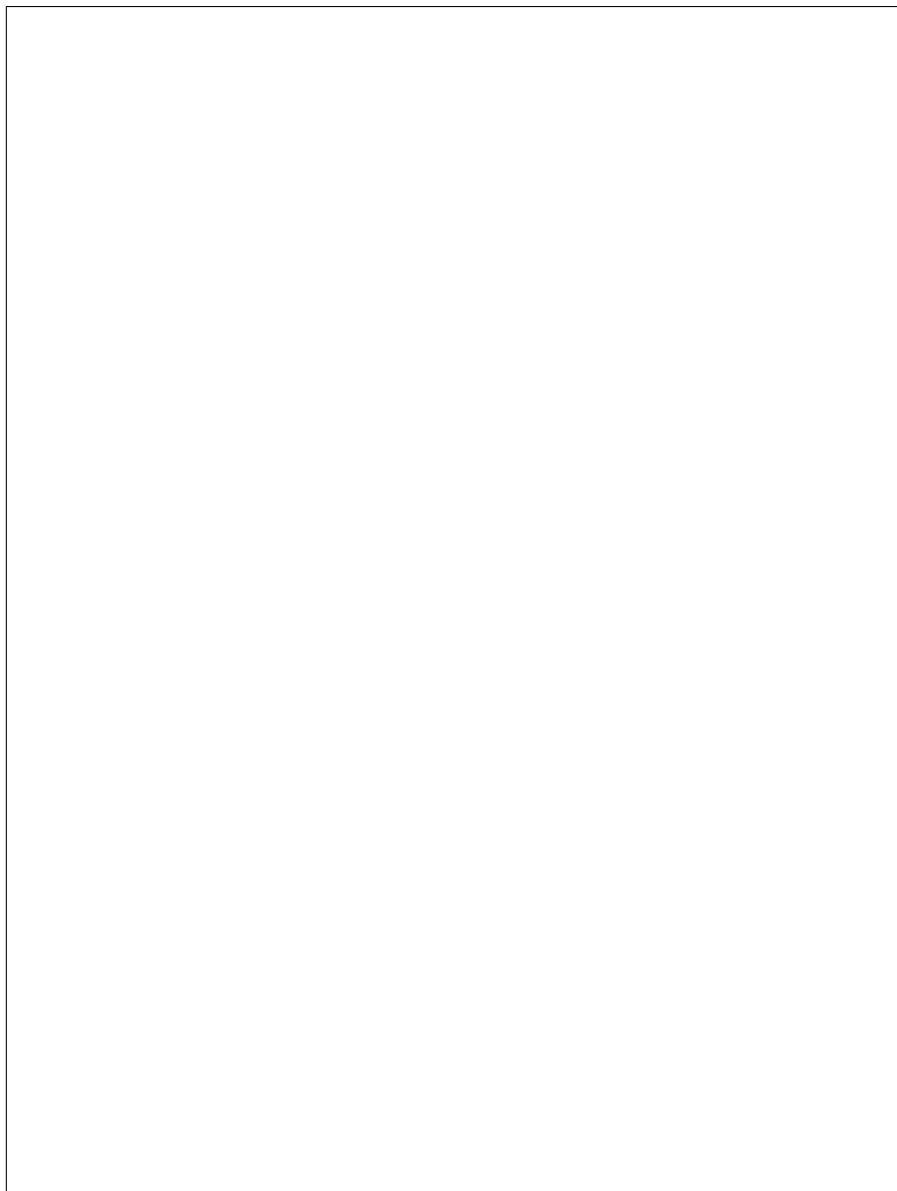
List
of
IP
ad-
dres
as-
so-
ci-
ated
with
task
port
and
port
grou

update_c

Send
or
up-
date
the
DHCP
BOC
op-
tions
for
this
node

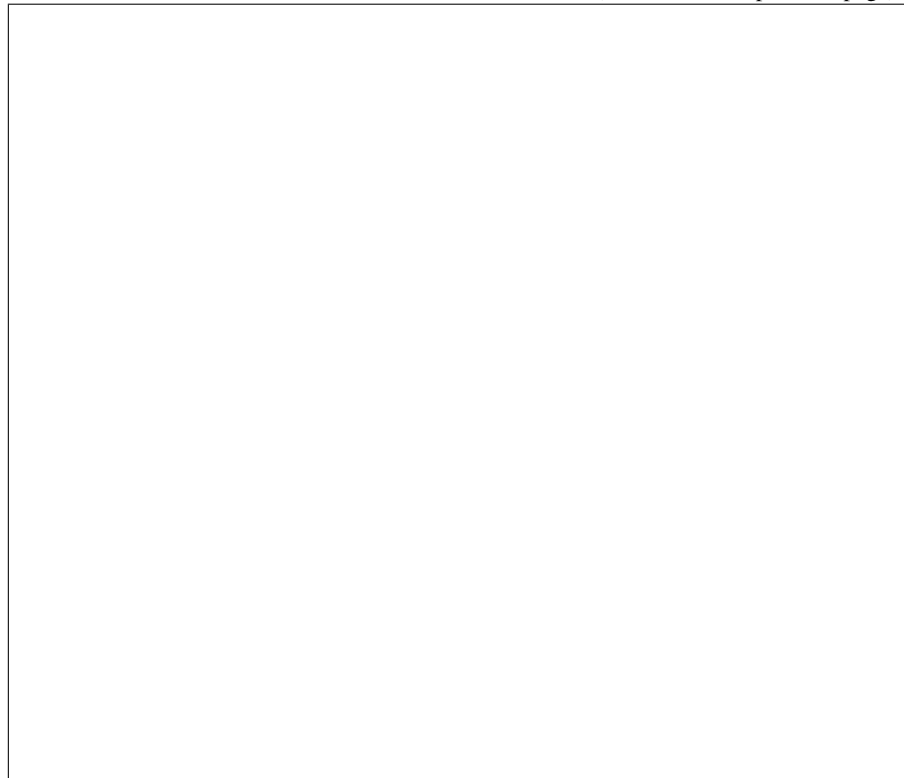
Parame

- **task**
A
Task
ager
in-
stan
- **opt.**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



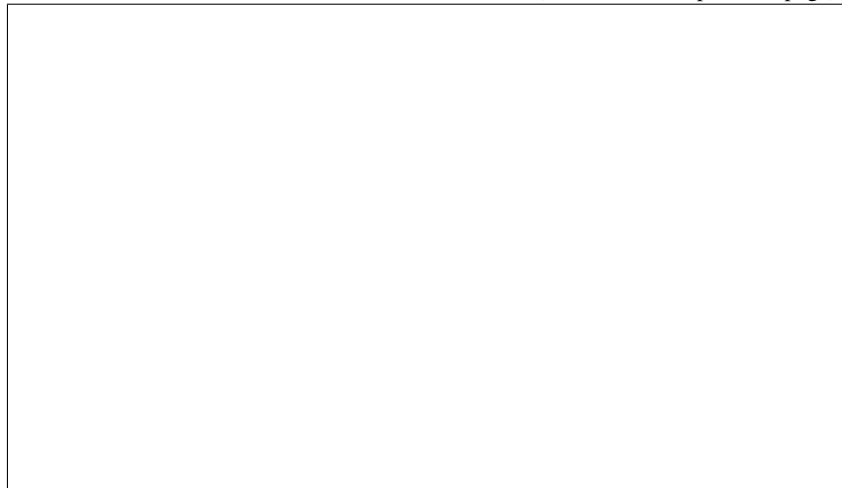
• **vif:**
A
dict
with
keys
port
and
port
grou
and
dicts
as
val-
ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port.
ob-
jects

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

update_p

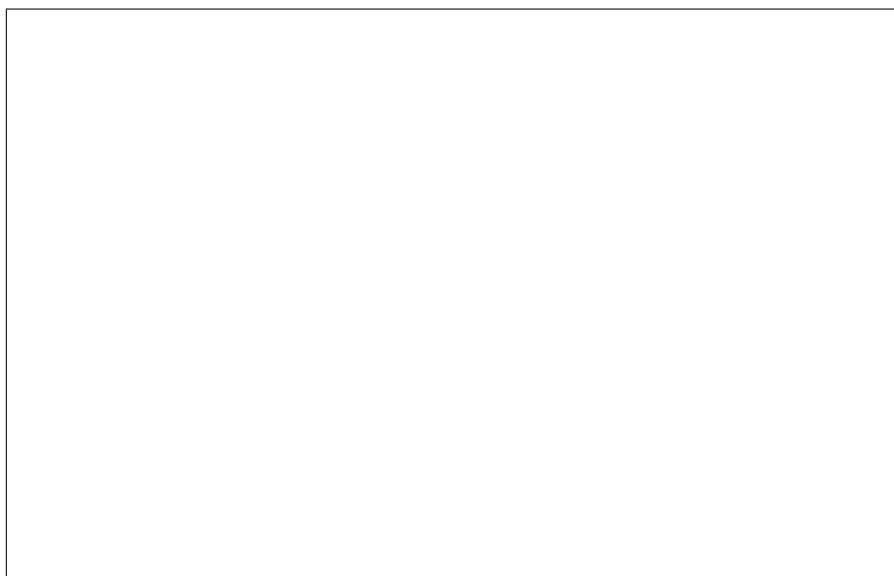
Up-
date
one
or
mor
DH
op-
tion

on
the
spec
i-
fied
port

Parame

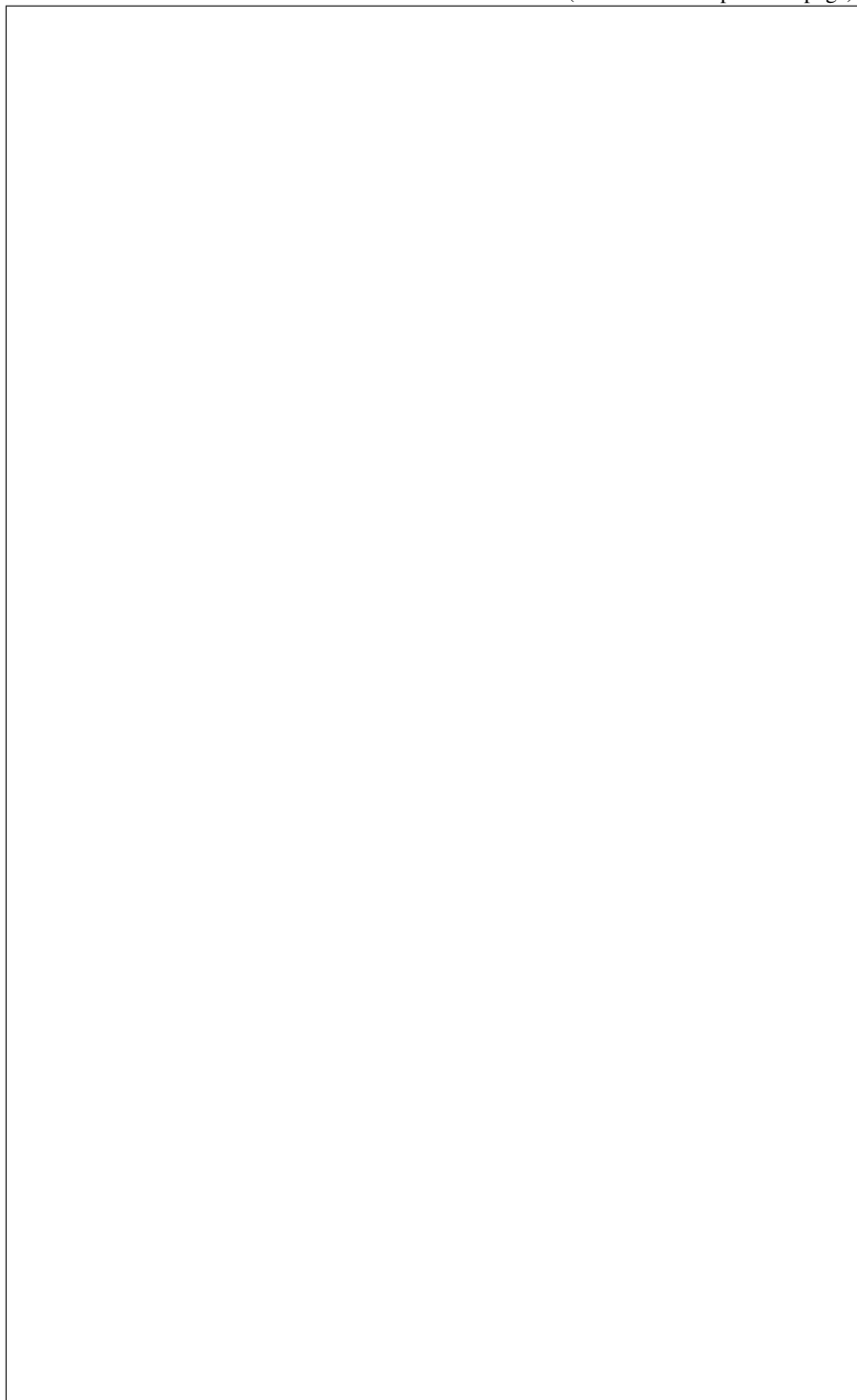
- **por-**
des-
ig-
nate
whic
port
thes
at-
tribu
will
be
ap-
plic
to.

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



- **tok**
An
op-
tion:
au-
then-
ti-
ca-

tion
to-
ken.
Dep
re-
cate
use
con-
text

- **con**
(irc
com
con
Req
re-
ques
con-
text

Raises
Fail
ToU
dat-
eD-
HCF
tOn-
Port

Module contents

`ironic.drivers` package

Subpackages

`ironic.drivers.modules` package

Subpackages

`ironic.drivers.modules.ansible` package

Submodules

`ironic.drivers.modules.ansible.deploy` module

An-
si-
ble

de-
ploy
in-
ter-
face

class ir

Base
iron
drive
model
age
Head
iron
drive
model
age
Age
iron
drive
base
Dep

In-
ter-
face
for
depl
relat
ac-
tions

clean_up

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

deploy(A

Per-

form
a
de-
ploy
men
to
a
node

execute_

Ex-
e-
cute
a
clear
step

Paramete

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

- **step**
a
clear
step
dic-
tio-
nary
to
ex-
e-
cute

Returns

Non

get_clea

Get
the
list

of
clear
step
from
the
file.

Parameter

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns

A
list
of
clear
step
dic-
tio-
nar-
ies

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

in_core

Che
if
we
are
in
the
de-

ploy
de-
ploy
step

As-
sum
that
we
are
in
the
DE-
PLC
WA
state

Paramete

task
a
Task
ager
in-
stan

Returns

True
if
the
cur-
rent
de-
ploy
step
is
de-
ploy

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

prepare_

Boo
into
the
rame
to
pre-
pare
for
clea
ing.

**Paramete
task**

a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Nod
if
the
pre-
vi-
ous
clea
ing
port
can-
not
be
re-
mov
or
if
new
clea
ing

ports cannot be created

Returns

Non
or
state
for

asyn
pre-
pare

process_

Star
the
next
clea
step
if
the
pre-
vi-
ous
one
is
com
plete

Parame

- **task**
a
Task
ager
in-
stan
- **step**
clea
or
de-
ploy

take_ove

Take
over
man
age-
men
of
this
task
node
from
a
deac
con-

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

tftpboot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

duc-
tor.
If
con-
duc-
tors
host
main-
tain
a
stati-
re-
la-
tion-
ship
to
node
this
meth-
shou-
be
im-

For exam
Neu-
tron
mus-
for-
war-
DHCP
BOC
re-
ques-
to
a
con-
duc-
tor
whic-
has
pre-
pare
the

Parame
tas!

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

tear_down

Tear
down
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node

tear_down

A
de-
ploy
step
to
tear
down
the
ager

Shut
down
the
ma-
chin
and
re-
mov
it
from
the

pro-
vi-
sion
ing
net-
work

Parameter

task

a
Task
ager
ob-
ject
con-
tain-
ing
the
node

tear_down

Clea
up
the
PXE
and
DHCP
files
af-
ter
clea
ing.

Parameter

task

a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

NodeError
if
the
clea
ing

port
can-
not
be
re-
mov

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

write_in

exceptio

Base
iron
exc
Iron

Module contents

`ironic.drivers.modules.drac` package

Submodules

`ironic.drivers.modules.drac.bios` module

DRA
BIO
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

class ir

Base
iron
dri
mod
red
bio
Red

iDR
Red-
fish
in-
ter-
face
for
BIO
setti
relat
ac-
tions

Pres
this
class
en-
tirely
de-
fers
to
its
base
class
a
gene
vend
inde
Red-
fish
in-
ter-
face

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

class ir

Base
iron
dri
bas
BIO

BIO
In-
ter-
face
Im-
ple-
men-
ta-
tion
for
iDR

apply_co

Ap-
ply
the
BIO
con-
fig-
u-
ra-
tion
to
the
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on
- **set**
List
of
BIO
set-

tings
to
ap-
ply

Raises

DRA
C-
Op-
er-
a-
tionl
upon
an
er-
ror
from
pyth
drac

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

or None if it is completed.

cache_b:

Stor
or
up-
date
the
cur-
rent
BIO
set-

tings
for
the
node

Get
the
cur-
rent
BIO
set-
ting
and
store
them
in
the
bios
data
ta-
ble.

Parame

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

factory_

Re-
set
the
BIO
set-
ting
of
the
node
to
the
fac-
tory
de-
fault

This
uses
the
Life
cy-
cle
Con-
troll
con-
fig-
u-
ra-
tion
to
per-
form
BIO
con-
fig-
u-

ration reset. Leveraging the python-dracclient methods already available.

Parame

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node

to
act
on

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
re-
set
is
in
prog
asyn
chro
or
Non
if

it is completed.

get_prop

Re-
turn
the
prop
er-
ties
of
the
BIO
In-

ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
<pro
erty
de-
scrip
tion:
en-
tries

validate

Val-
i-
date
the
driv
spec
in-
for-
ma-
tion
used
by
the
idra
BM

Paramet

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on

the node or on invalid inputs

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on

ironic.d

Aba
dons
un-
com
mit-
ted
char
add
by
set_

Paramet

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Dra-

c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

Com
mits
penc
ing
char
add
by
set_

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **rebo**
in-
di-
cate
whe
a
re-
boot
job

the config job.

shou
be
au-
to-
mat-
i-
cally
cre-
ated
with

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

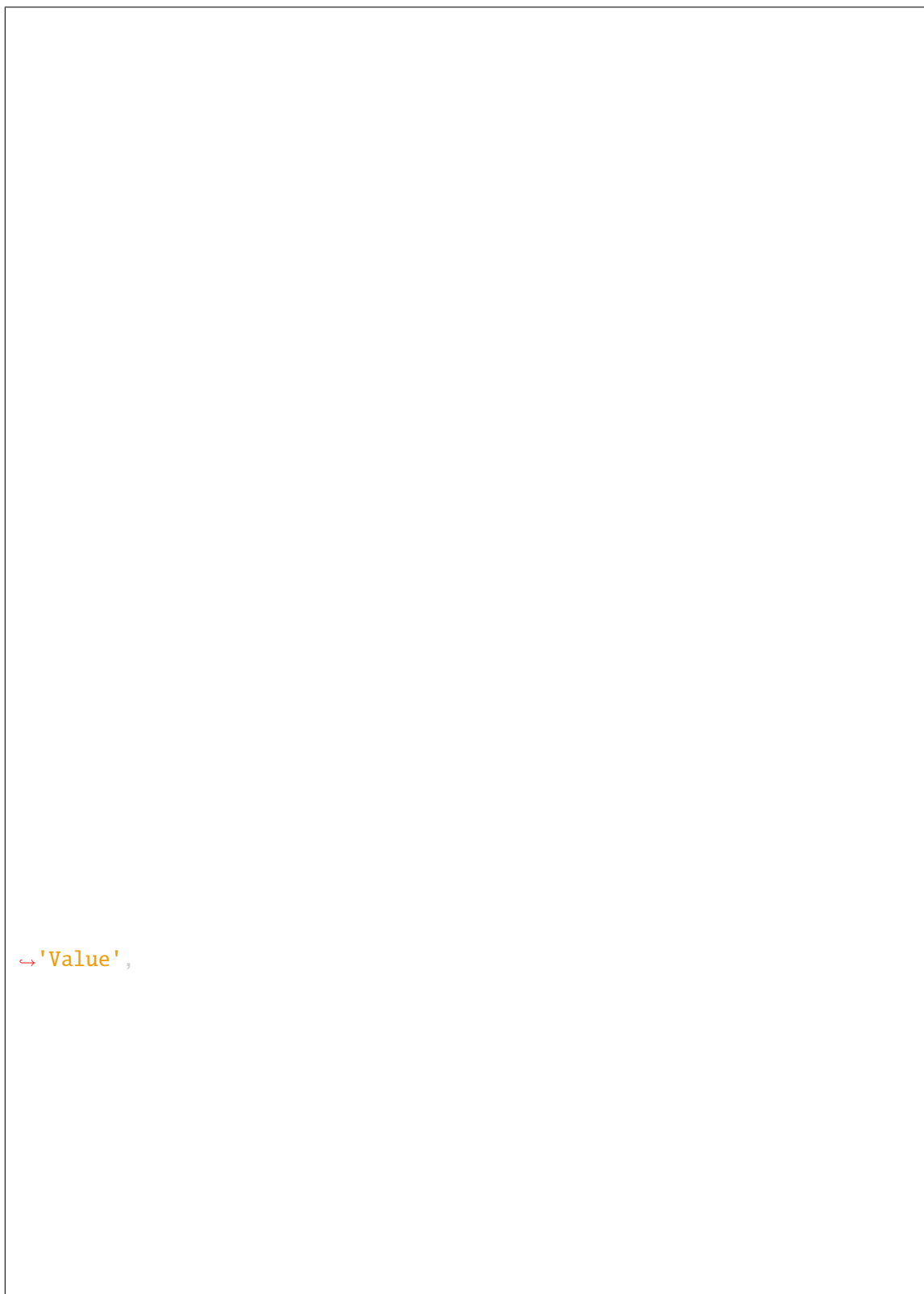
Returns

the
job_
key
with
the
id
of
the
new
cre-
ated
con-
fig
job.

ironic.d

Get
the
BIO
con-
fig-
u-
ra-
tion.

The
BIO
set-
ting
look
like:



```
↔ 'Value',
```

(continues on next page)

(continued from previous page)

```
↔False,
```

```
↔['Value', 'New Value', 'None']},
```

(continues on next page)

(continued from previous page)

```
↔value': 'Information',
```

```
↔value': None,
```

(continues on next page)

(continued from previous page)

```
↔length': 0,
```

```
↔length': 255,
```

(continues on next page)

(continued from previous page)

```
↔ 'current_value': 0,
```

(continues on next page)

```
↔ 'pending_value': None,
```

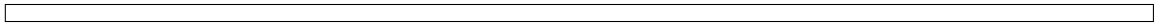
(continued from previous page)

```
↔ 'read_only': True,
```

```
↔ 'lower_bound': 0,
```

(continues on next page)

(continued from previous page)



Parameter
node
an
iron
node
ob-
ject.

Raises
Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns
a
dic-
tio-
nary
con-
tain-
ing
BIO
set-
tings

The
abov
val-
ues
are
only
ex-
am-
ples
of
cour
BIO
at-
tribu

ex-
pose
via
this
API
will

always be either an enumerated attribute, a string attribute, or an integer attribute. All attributes have the following parameters:

Parameter

- **name**
is
the
name
of
the
BIO
at-
tribu

- **cur:**
is
the
cur-
rent
valu
of
the
at-
tribu
It
will
al-
ways
be
ei-
ther
an
in-

teger or a string.

- **pen**
is
the
new
valu
that
we

pending value.

will result in an error. The read-only flag can change depending on other attributes. A future version of this call may expose the dependencies that indicate when that may happen.

wan
the
at-
tribu
to
have
Non
mea
that
there
is
no

- **rea**
in-
di-
cate
whe
this
at-
tribu
can
be
chan
Try-
ing
to
chan
a
read
only
valu

Enu
mer-
able
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

to.

Parameter
pos
is
an
ar-
ray
of
val-
ues
it
is
per-
mis-
si-
ble
to
set
the
at-
tribu

Strin
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

Parameter

- **min**
is
the
min-
i-
mun
leng
of
the
strin
- **max**

It may be None if the string is read only or if the string does not have to match any particular regular expression.

is
the
max
i-
mun
leng
of
the
strin

- **pcr**
is
a
PCR
com
pat-
i-
ble
reg-
u-
lar
ex-
pres
sion
that
the
strin
mus
matc

In-
te-
ger
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

Paramet

-

low
is
the
min-
i-
mun
valu
the
at-
tribu
can
have

- **upp**
is
the
max
i-
mun
valu
the
at-
tribu
can
have

`ironic.d`

Sets
the
pend
ing_
pa-
ram-
e-
ter
for
each
of
the
val-
ues
pass
in.

Paramet

- **task**
a
Task

ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **kwargs**
a
dic-
tio-
nary
of
{At-
tribu
Nam
New
Valu

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
is_c
key
with
a

`commit_config()` needs to be called to make the changes, and the `is_reboot_required` key which has a value of true or false. This key is used to indicate to the `commit_config()` call if a reboot should be performed.

`ironic.drivers.modules.drac.boot` module

bool
valu
in-
di-
cat-
ing
whe

class ir

Base
iro
dri
mod
red
boo
Red

iDR
Red-
fish
in-
ter-
face
for
vir-
tual
me-
dia
boot
relat
ac-
tions

Vir-
tual
Me-
dia
al-
lows
boot
ing
the
sys-
tem
from

image that BMC inserts into the drive.

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

vir-
tual
CD/
drive
con-
tain-
ing
user

The
CD/
im-
ages
must
be
in
ISO
for-
mat
and
(de-
pend-
ing
on
BMC
im-
ple-
men-
ta-

The
base
line
boot
work-
flow
is
most
base
on
the
stan-
dard
Red-
fish
vir-
tual
me-
dia
boot

interface, which looks like this:

EFI boot loader) images

Swift temporary URL

1. Pull kernel ramdisk and ESP if UEFI boot is requested (FAT partition image with

2. Create boot ISO out of images (#1) push it to Glance and pass to the BMC as

3. Optionally create floppy

push it to Glance and pass to the BMC as Swift temporary URL

im-
age
with
de-
sired
sys-
tem
con-
fig-
u-
ra-
tion
data

4.
In-
sert
CD/
and
(op-
tion-
ally)
flopp
im-
ages
and
set
prop
boot
mod

For
buil
ing
de-
ploy
or
res-
cue
ISO
red-
fish
boot
in-
ter-
face
uses
*de-
ploy*
or
res-

cue_kernel/rescue_ramdisk properties from *[instance_info]* or *[driver_info]*.

in the Glance image metadata found in *[instance_info]image_source* node property.

to boot from a virtual media device - this is done via OEM action call implemented in Dell sushy OEM extension package.

For
buil
ing
boot
(use
ISO
red-
fish
boot
in-
ter-
face
seek
*ker-
nel_
and
ram
prop
er-
ties*

iDR
vir-
tual
me-
dia
boot
in-
ter-
face
only
dif-
fers
by
the
way
how
it
sets
the
node

ironic.drivers.modules.drac.common module

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
DRA
mod
ules

ironic.d

Re-
turn
a
DRA
Clie
ob-
ject
from
pyth
drac
li-
brar

Parameter
node
an
iron
node
ob-
ject.

Returns
a
DRA
Clie
ob-
ject.

Raises
In-
valid

node or on invalid input.

combination of both.

Pa-
ram-
e-
ter-
Valu
if
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the

ironic.d

Pars
a
node
driv
val-
ues.

Pars
the
driv
of
the
node
read
de-
fault
val-
ues
and
re-
turn
a
dict
con-
tain-
ing
the

Paramet
nod
an

iron
node
ob-
ject.

Returns

a
dict
con-
tain-
ing
in-
for-
ma-
tion
from
driv
and
de-
fault
val-
ues.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on

the node or on invalid inputs.

ironic.drivers.modules.drac.inspect module

DRA
in-
spec
tion
in-
ter-
face

class ir

Base
iro
dri
mod
dra
ins
Dra

Clas
alias
of
class
Drac
Man
In-
spec

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idra
in-
spec
in-
ter-
face

implementation entrypoint.

All

That makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Dra
Man
In-
spec

class ir

Base
iro
dri
mod
red
ins
Red

iDR
Red-
fish
in-
ter-
face
for
insp
relat
ac-
tion:

inspect

In-
spec
hard
ware
to
get

essential properties are not received from the node.

the
hard
ware
prop
er-
ties.
In-
spec
hard
ware
to
get
the
es-
sen-
tial
prop
er-
ties.
It
fails
if
any
of
the
es-

Parame

task
a
Task
ager
in-
stan

Raises

Har
ware
spec
tion-
Fail-
ure
if
es-
sen-
tial
prop
er-
ties
coul
not

cessfully.

be
re-
triev
suc-

Returns

The
re-
sult-
ing
state
of
in-
spec
tion.

class ir

Base
iro
dri
bas
Ins

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

inspect_

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tion
hard
ware
prop
er-
ties.

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Har
ware
spec
tion-
Fail-
ure,
if
un-

able
to
get
es-
sen-
tial
hard
ware
prop
er-
ties.

Returns
state

validate

Val-
i-
date
the
drive
spec
info
sup-
plied
This
meth
val-
i-
date
when
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-
quir

information for this driver to manage the node.

Paramet
task
a
Task

ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid

on the node.

ironic.drivers.modules.drac.job module

DRA
Life
cy-
cle
job
spe-
cific
meth
ods

ironic.d

Get
the
de-

tails
of
a
Life
cy-
cle
job
of
the
node

Paramet

- **node**
an
iron
node
ob-
ject.
- **job**
ID
of
the
Life
cy-
cle
job.

Returns

a
Job
ob-
ject
from
drac
clien

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth

drac
ironic.d
List
un-
fin-
ishe
con-
fig
jobs
of
the
node

Paramet
nod
an
iron
node
ob-
ject.

Returns
a
list
of
Job
ob-
jects
from
drac
clien

Raises
Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

ironic.d
Val-
i-
date

node: an ironic node object. :param name_prefix: A name prefix for jobs to validate. :raises: DracOperationError on an error from python-dracclient.

the
job
queu
on
the
node

It
raise
an
ex-
cep-
tion
if
an
un-
fin-
ishe
con-
fig-
u-
ra-
tion
job
ex-
ists.
:par

ironic.d

Wait
for
job
to
com
plete

It
will
wait
for
the
job
to
com
plete
for
20
min-
utes
and

complete within given interval of time. :param node: an ironic node object. :param retries: no of retries to make conductor wait. :raises: DracOperationError on exception raised from python-dracclient or a timeout while waiting for job completion.

ironic.drivers.modules.drac.management module

raise
time
out
if
job
never

DRAC
man
age-
men
in-
ter-
face

class ironic.drivers.modules.drac.management.DracManagement

Base
ironic
drivers
modules
drac
management
DracManagement

Class
alias
of
class:
DracManagement
Management

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the

face implementation endpoint.

ment. That makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

dep-
re-
cate
idrac
man
age-
men
in-
ter-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Dra
Man
Man
age-

class ir

Base
iro
dri
mod
red
man
Red

iDR
Red-
fish
in-
ter-
face
for
man
relat

ac-
tions

```
EXPORT_C  
{'descr:  
'require
```

```
IMPORT_C  
{'descr:  
'require
```

```
IMPORT_I  
{'descr:  
'require  
'URL of  
True}}
```

clear_jobs

Clea
iDR
job
queu

Parameter

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Red
fish-
Er-
ror
on
an
er-
ror.

export_c

Ex-

in specific format in indicated location.

port
the
con-
fig-
u-
ra-
tion
of
the
serv

Ex-
port
the
con-
fig-
u-
ra-
tion
of
the
serv
agai
whic
the
step
is
run
and
store
it

Uses
Dell
Serv
Con
fig-
u-
ra-
tion
Pro-
file
(SCI
from
sush
oem
idra
li-
brar
to
get

configuration for cloning.

ALL

Parameter

- **task**
A task from Task Manager.
- **export_uri**
URI of location to save the configuration to.

Raises

MissingParameterError if configuration name of a file

to save the configuration to

Raises

configuration export from SCP failed

Dra-
c-
Op-
er-
ata-
tion-
Erro
whe
no
man
a-
gage
for
Red-
fish
sys-
tem
foun
or

Raises

Red-
fish-
Er-
ror
whe
load
ing
OEM
ex-
ten-
sion
faile

import_

Im-
port
and
ap-
ply
the
con-
fig-
u-
ra-
tion
to
the
serv
Gets

into given server. Uses Dells Server Configuration Profile (SCP).

pre-
crea
con-
fig-
u-
ra-
tion
from
stor-
age
by
give
lo-
ca-
tion
and
im-
port
that

Parame

- **task**
A
task
from
Task
ager

- **imp**
URI
of
lo-
ca-
tion
to
fetc
de-
sirec
con-
fig-
u-
ra-
tion
from

Raises
Miss
ing-

to fetch the configuration from

Pa-
ram-
e-
ter-
Valu
if
miss
ing
con-
fig-
u-
ra-
tion
nam
of
a
file

import_

Im-
port
and
ex-
port
con-
fig-
u-
ra-
tion
in
one
go.

Gets
pre-
crea
con-
fig-
u-
ra-
tion
from
stor-
age
by
give
nam
and
im-
port

server. After that exports the configuration of the server against which the step is run and stores it in specific format in indicated storage as configured by Ironic.

that
into
give

Parameters

- **import**
URI
of
lo-
ca-
tion
to
fetch
de-
sired
con-
fig-
u-
ra-
tion
from
- **export**
URI
of
lo-
ca-
tion
to
save
the
con-
fig-
u-
ra-
tion
to.

known_good_state

Re-
set
iDR
to
know
good
state

queue.

An
iDR
is
re-
set
to
a
know
good
state
by
re-
set-
ting
it
and
clear
ing
its
job

Paramete

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Red-
fish-
Er-
ror
on
an
er-
ror.

reset_i

Re-
set
the

iDR

Parame

tasl

a

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Red-

fish-

Er-

ror

on

an

er-

ror.

class ir

Base

iro

dri

bas

Man

clear_jc

Clea

the

job

que

Parame

tasl

a

Task

ager

in-

stan

con-

tain-

ing

the
node
to
act
on.

Returns

Non
if
it
is
com
plete

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Parameters

task

a

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Raises

Dra-

c-

Op-

er-

a-

tion)

on

an

er-

ror

from

pyth

drac

Returns

a

dic-

tio-

nary

con-

tain-

ing:

boot_c

the

boot

de-

vice

one

of

iron

com

boo

or

unknown.

Non
if
it
is
un-
know

persist

wh
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

get_sens

Get
sen-
sors
data

Paramet

task
a
Task

ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data
fails

Raises

Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Returns

re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type

can be processed by Ceilometer.

whic

get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
iro
com
boo

known_g

Re-
set
the

iDR
Clea
the
job
que

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

Non
if
it
is
com
plete

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

reset_ic

Re-
set
the
iDR

Parame

tas

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

Non-
if
it
is
com-
plete

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice

to
use
on
next
re-
boot
of
the
node

Paramete

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **dev**
the
boot
de-
vice
one
of
iron
com
boo
- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-

not. Default: False.

sist
to
all
fu-
ture
boot
Fals
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

validate

Val-
i-
date
the
drive
spec
info
sup-
plied

This
meth
val-
i-
date
whe
the
drive
prop
erty
of

information for this driver to manage the node.

on the node.

the
sup-
plied
node
con-
tains
the
re-
quir

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid

ironic.d

Set
the

boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
boot
of
the
node

Parameter

- **node**
an
ironic
node
ob-
ject.
- **dev:**
the
boot
de-
vice
one
of
ironic
command
boot
- **per:**
Boo
valu
True
if
the
boot
de-
vice

not. Default: False.

ironic.drivers.modules.drac.power module

will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

DRA
pow
in-
ter-
face

class ir

Base
ironic
drivers
modules
drac
power
Drac
Class
alias
of
class
Drac
Man

plementation entrypoint.

makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

Pow
This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idrac
pow
in-
ter-
face
im-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
DraC
Man
Pow
That

class ir

Base
iro
dri
mod

red
pow
Red

iDR
Red
fish
in-
ter-
face
for
pow
relat
ac-
tion

Pres
this
class
en-
tirely
de-
fers
to
its
base
class
a
gene
vend
inde
Red
fish
in-
ter-
face

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

class ir

Base
iro
dri
bas
Pow

In-
ter-
face
for
pow
relat

ac-
tions

get_pow

Re-
turn
the
pow
state
of
the
node

Paramete

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

the
pow
state
one
of
iron
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
DRA

cre-
den-
tials
are
miss-
ing.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

reboot()

Per-
form
a
re-
boot
of
the
task
node

Parame

- **task**
a

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises
In-
valid
Pa-
ram-
e-
ter
Valu
if
re-
quir
DRA
cre-
den-
tials
are
miss
ing.

Raises
Dra-
c-
Op-
er-

a-
tion
on
an
er-
ror
from
pyth
drac

set_power

Set
the
power
state
of
the
node

Parameters

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **power**
a
power
state
from
[ironic-compute](#)
[state](#)
- **time**
time
out
(in

sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
DRA
cre-
den-
tials
are
miss
ing.

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

validate

Val-
i-
date
the
driv
spec

information for this driver to manage the power state of the node.

Nod
pow
info.

This
meth
val-
i-
date
whe
the
driv
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valie
Pa-
ram-
e-
ter-
Valu
if
re-
quir

on the node.

ironic.drivers.modules.drac.raid module

drive
at-
tribu
is
miss
ing
or
in-
valid

DRA
RAI
spe-
cific
meth
ods

class ir

Base
iro
dri
mod
dra
rai
Dra

Clas
alias
of
class
Dra
Man
RAI

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-

plementation entrypoint.

makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

cate
idrac
RAI
in-
ter-
face
im-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Drac
Man
RAI
That

class ir

Base
iron
dri
mod
red
rai
Red

iDR
Red-
fish
in-
ter-
face
for
RAI
re-
latec
ac-

tions
In-
clud
iDR
spe-
cific
ad-
just-
men
for
RAI
re-
latec
ac-
tion:

create_c

Cre-
ate
RAI
con-
fig-
u-
ra-
tion
on
the
node

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
as
read
from
node
This
meth
by
de-
fault

will create all logical disks.

Parameter

- **task**
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **create**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

ified in the nodes `target_raid_config`. Default value is `True`.

- **create**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
non-
root

cept the root volume) in the nodes target_raid_config. Default value is True.

creating the new configuration. Default is False.

plete.

vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

Raises

Red-
fish-

Er-
ror
if
there
is
an
er-
ror
cre-
at-
ing
the
con-
fig-
u-
ra-
tion

delete_c

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
node

Parame

task
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if

it is complete.

dele
tion
is
in
prog
asyn
chro
or
Non
if

class ir

Base
iro
dri
bas
RAI

apply_co

Ap-
plies
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

ra-
tion
to
ap-
ply.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-
- **del**
Set-

creating the new configuration.

ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn

plete.

chro
or
Non
if
it
is
com

create_c

Cre-
ate
the
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

erwise, no root volume is created. Default is True.

ated. Default is True.

act
on.

- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

- **del**
Set-
ting
this
to

creating the new configuration. Default is False.

it is completed.

True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
cre-
ation
is
in
prog
asyn
chro
or
Non
if

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or

emp

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

delete_c

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

state
(clea
ing)
or
state
(de-
ploy
men

it is completed.

if
dele
tion
is
in
prog
asyn
chro
or
Non
if

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

get_log:

Get
the
RAI
con-
fig-
u-
ra-
tion
of
the
node

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the

node
to
act
on.

Returns

A
dic-
tio-
nary
of
prop
er-
ties.

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

ironic.d

Dele
all
pen
ing
char
on
a
RAI

con-
troll

Parameter

- **node**
an
ironic
node
ob-
ject.
- **raid**
id
of
the
RAID
con-
troll

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

Con
vert
disk
RAID
sta-
tus

This
meth
con-
verts
the
re-
ques

this by only converting the disks that are not already in the correct state.

phys
i-
cal
disk
from
RAI
to
JBO
or
vice
vers
It
does

Parameter

- **node**
an
iron
node
ob-
ject.
- **mode**
the
mod
to
chan
the
disk
ei-
ther
to
RAI
or
JBO
- **con**
Dic-
tio-
nary
of
con-
troll
and
cor-
re-
spor

requested mode.

troller ids to the conversion results for that controller. The conversion results are a dict that contains:

- The `is_commit_required` key with the value always set to `True` indicating that a config job must be created to complete disk conversion.
- The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted to complete disk conversion.

ing
disk
ids
to
con-
vert
to
the

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

con-
ver-
sion
a
dic-
tio-
nary
that
map
con-

Raises

DRA
C-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

Free

up
the
for-
eign
drive

Parameter

- **node**
an
iron
node
ob-
ject.
- **raid**
id
of
the
RAID
con-
troll

Returns

a
dic-
tio-
nary
con-
tain-
ing
-

The
is_c
need
key
with
a
bool
valu
in-
di-

cating whether a config job must be created for the values to be applied. - The is_reboot_required key with a RebootRequired enumerated value indicating whether the server must be rebooted to clear foreign configuration.

Raises

Dra-
c-
Op-

er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

Ap-
ply
all
pend
ing
char
on
a
RAI
con-
troll

Paramet

- **node**
an
iron
node
ob-
ject.
- **raid**
id
of
the
RAI
con-
troll
- **reboot**
in-
di-
cate
whe
a
re-
boot

the config job. (optional, defaults to False)

job
shou
be
au-
to-
mat-
i-
cally
cre-
ated
with

- **rea.**
in-
di-
cate
RAI
con-
troll
sup-
port
re-
al-
time
(op-
tiona
de-
fault
to
Fals

Returns
id
of
the
cre-
ated
job

Raises
Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from

pyth
drac
ironic.d

Cre-
ate
a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

The
cre-
ated
vir-
tual
disk
will
be
in
pend-
ing
state
The
DRA
card
will
do
the
ac-
tual

configuration once the changes are applied by calling the `commit_config` method.

Parameter

- **node**
an
iron
node
ob-

ject.

- **raid**
id
of
the
RAID
con-
troll

- **phys**
ids
of
the
phys
i-
cal
disk

- **raid**
RAID
leve
of
the
vir-
tual
disk

- **size**
size
of
the
vir-
tual
disk

- **dis**
nam
of
the
vir-
tual
disk
(op-
tiona

- **spa**
Num

ber
of
span
in
vir-
tual
disk
(op-
tiona

- **span**
Num-
ber
of
disk
per
span
(op-
tiona

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
com
mit_
key
with
a
bool
valu
in-
di-
cat-
ing

whether a config job must be created for the values to be applied.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-

ror
from
pyth
drac

ironic.d

Dele
a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

The
dele
vir-
tual
disk
will
be
in
pend
ing
state
The
DRA
card
will
do
the
ac-
tual
con-

figuration once the changes are applied by calling the `commit_config` method.

Parameter

- **node**
an
iron
node
ob-
ject.
-

vir-
id
of
the
vir-
tual
disk

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
com
mit_
key
with
a
bool
valu
in-
di-
cat-
ing

whether a config job must be created for the values to be applied.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

ironic.d

List
the
phys
i-
cal
disk

of
the
node

Parameter

node
an
ironic
node
ob-
ject.

Returns

a
list
of
Physical
iscalD
isk
ob-
jects
from
drac
client

Raises

Drac-
c-
Oper-
er-
a-
tion
on
an
er-
ror
from
python
drac

ironic.d

List
the
RAI
con-
troll
of
the
node

Parameter
node

an
iron
node
ob-
ject.

Returns

a
list
of
RAI
Con
troll
ob-
jects
from
drac
clien

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

List
the
RAI
con-
fig-
u-
ra-
tion
set-
tings

Paramet

node
an
iron
node
ob-

tributes are RAIDEnumerableAttribute, RAIDStringAttribute and RAIDIntegerAttribute objects.

ject.
Returns
a
dic-
tio-
nary
with
the
RAI
set-
ting
us-
ing
In-
stan-
ceID
as
the
key.
The
at-

Raises
DRA
C-
Op-
er-
a-
tionl
on
er-
ror
re-
port
back
by
the
DRA
in-
ter-
face

ironic.d

List
the
vir-
tual
disk
of
the

node

Parameter

node

an
ironic
node
ob-
ject.

Returns

a
list
of
Vir-
tuall
isk
ob-
jects
from
drac
client

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.d

Sets
the
RAI
con-
fig-
u-
ra-
tion

It
sets
the
pen-
ing_

ues to be applied, a config job must be created.

pa-
ram-
e-
ter
for
each
of
the
at-
tribu-
pass
in.
For
the
val-

Parameter

- **node**
an
iron
node
ob-
ject.
- **con**
the
ID
of
the
RAI
con-
troll
- **set**
a
dic-
tio-
nary
con-
tain-
ing
the
pro-
pose
val-
ues,
with

name of attribute and the value being the proposed value.

ing whether a config job must be created for the values to be applied. - The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted for the values to be applied. Possible values are `true` and `false`.

each
key
be-
ing
the

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

The
is_c
key
with
a
bool
valu
in-
di-
cat-

Raises

DRA
C-
Op-
er-
a-
tionl
on
er-
ror
re-
port
back
by
the
DRA
in-
ter-
face

ironic.drivers.modules.drac.utils module

ironic.d

Load
OEM
man
ager
and
ex-
e-
cute
pass
meth
on
it.

Know
iDR
Red-
fish
sys-
tems
has
only
one
man
ager
but
as
Red-
fish
sche
al-
lows
a
list

this method iterates through all values in case this changes in future. If there are several managers, this will try starting from the first in the list until the first success.

Paramet

- **task**
a
Task
ager
in-
stan
-

sages.

tension manager. Example: lambda m: m.reset_idrac() For older versions also support second input parameter Redfish manager itself when pass_manager set to True.

pro
user
frier
nam
of
meth
to
be
ex-
e-
cute
Used
in
ex-
cep-
tion
and
log
mes

- **lamb**
meth
to
ex-
e-
cute
as
lamb
func
tion
with
in-
put
pa-
ram-
e-
ter
OEM
ex-

- **pas**
whe
to
pass
man
ager
it-

for backward compability, new functions must not pass manager, but acquire it internally. Will be removed in future.

self
to
ex-
e-
cute
OEM
ex-
ten-
sion
meth
This
is

Returns
Re-
turn
valu
of
lamb

Raises
Red-
fish-
Er-
ror
if
cant
ex-
e-
cute
OEM
func
tion
ei-
ther
be-
caus
ther
are
no

managers to the system, failed to load OEM extension or execution of the OEM method failed itself.

ironic.drivers.modules.drac.vendor_passthru module

DRAC
vend
pass
in-
ter-
face

class `ironic.drivers.modules.drac.vendor_passthru`

Base
ironic.drivers.modules.drac.vendor_passthru
Redfish

iDRAC
Redfish
fish
in-
ter-
face
for
ven-
dor_

Use
the
Redfish
fish
im-
ple-
men-
ta-
tion
for
ven-
dor
pass

class `ironic.drivers.modules.drac.vendor_passthru`

Base
ironic.drivers.modules.drac.vendor_passthru
DRAC

face implementation entrypoint.

Class
alias
of
class:
Drac
Man
Ven-
dor-
Pass
This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idra-
ven-
dor
pass
in-
ter-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class:
Drac
Man
Ven-
dor-

Passthru. That makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

class ir

Base
iro
dri
bas
Ven

In-
ter-
face
for
DRA
spe-
cific
meth
ods.

abandon

Aba
don
a
BIO
con-
fig-
u-
ra-
tion
job.

This
meth
is
used
to
aband
don
a
BIO
con-
fig-
u-
ra-
tion
pre-
vi-
ousl
sub-

through `set_bios_config()`.

mit-
ted

Parameters

- **task**
a Task Manager instance containing the node to act on.
- **kwargs**
not used

Raises

Drac c-Operation on an error from pyth drac

commit_l

Com
mit
a
BIO
con-
fig-
u-
ra-
tion

job.
This
meth
is
used
to
com
mit
a
BIO
con-
fig-
u-
ra-
tion
job.
sub-
mit-
ted
thro
set_

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **rebo**
in-
di-
cate
whe
a
re-
boot
job
shou

the config job.

fig job, and the `reboot_required` key indicating whether the node needs to be rebooted to start the config job.

be
au-
to-
mat-
i-
cally
cre-
ated
with

- **kwargs**
not
used

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
job.
key
with
the
id
of
the
new
cre-
ated
con-

get_bios

Get the BIOS configuration.

This method is used to retrieve the BIOS settings from a node.

Parameters

- **task**: a TaskManager instance containing the node to act on.
- **kwargs**: not used.

Raises

Drac-
c-
Op-

er-
a-
tion
on
an
er-
ror
from
pyth
drac

Returns

a
dic-
tio-
nary
con-
tain-
ing
BIO
set-
tings

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

list_un

List
un-
fin-
ishe
con-
fig
jobs
of
the
node

Parame

•

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **kwargs**
not
used

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
unf-
key;
this
key
point
to
a
list
of
dicts
with

each dict representing a Job object.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror

from
pyth
drac

set_bios

Cha
BIO
set-
ting

This
meth
is
used
to
chan
the
BIO
set-
ting
on
a
node

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **kwargs**
a
dic-
tio-
nary
of
{At-
tribu
Nam

New
Valu

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
is_
key
with
a
Boo
valu
in-
di-
cat-
ing
whe

commit_bios_config() needs to be called to make the changes, and the is_reboot_required key with a value of true or false. This key is used to indicate to the commit_bios_config() call if a reboot should be performed.

validate

Val-
i-
date
the
driv
spec
info
sup-
plied

information for this driver to manage the power state of the node.

This
meth
val-
i-
date
whe
the
driv
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **kwargs**
not
used

Raises

In-
valie
Pa-
ram-
e-
ter-

on the node.

Module contents

`ironic.drivers.modules.ibmc` package

Submodules

`ironic.drivers.modules.ibmc.management` module

Valu
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid

iBM
Man
age-
men
In-
ter-
face

class `ir`

Base
iro
dri
bas
Man

get_boot

Get
the
cur-
rent
boot
de-
vice
for

a
node

Parameters

task
A
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to

con-
nect
to
iBM

Raises

IBM
CER
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
iro
comm
boo
or
Non
if
it
is
un-
know

persist

Boo
valu
or

disabled.

Non
True
if
the
boot
de-
vice
per-
sists
Fals
oth-
er-
wise
Non
if
its

get_boot

Get
the
cur-
rent
boot
mod
for
a
node
Pro-
vide
the
cur-
rent
boot
mod
of
the
node

Paramete

task
A
task
from
Task
ager

Raises

In-
valid
Pa-

ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-

ror
in-
for-
ma-
tion

Returns

The
boot
mod
one
of
iron
com
boo
or
Non
if
it
is
un-
know

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sens

Get

sen-
sors
data

Not
im-
ple-
men
for
this
driv

Raises

NotI
ple-
men
ed-
Er-
ror

get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

task
a
task
from
Task
ager

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram

e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CER
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns

A
list

with
the
sup-
port
boot
de-
vice
de-
fined
in
iro
comm
boo

get_supp

Get
a
list
of
the
sup-
port
boot
mod

Parame

task
A
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
mod
de-
fined
in
iro
comm
boo
If
boot
mod
sup-

can't be determined, empty list is returned.

port

inject_1

In-
ject
NM
Non
Mas
able
In-
ter-
rupt
In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-

e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror

in-
for-
ma-
tion

set_boot

Set
the
boot
de-
vice
for
a
node

Paramete

- **task**
A
task
from
Task
ager
- **dev**
The
boot
de-
vice
one
of
iron
comm
boot
- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all

not. Default: False.

fu-
ture
boot
Fals
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CER-
ror
r
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Parameter

- **task**
A
task
from
Task
ager
- **mod**
The

boot
mod
one
of
iro
com
boo

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to

iBM

Raises

IBM
CER-
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

validate

Val-
i-
date
the
driv
in-
for-
ma-
tion
need
by
the
iBM
driv

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-

valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

ironic.drivers.modules.ibmcp.mappings module

iBM
and
Iron
con-
stan
map
ping

ironic.drivers.modules.ibmcp.power module

iBM
Pow
In-
ter-
face

class ir

Base
iron
dri
bas
Pow

get_powe

Get
the
cur-
rent
pow
state
of
the
task
node

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

A
pow
state
One
of
iron
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-

Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr-
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-

ma-
tion

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list
of
the
sup-
port
pow
state

Paramet

task
A
Task
ager
in-
stan-
con-
tain-
ing
the

at the moment.

node
to
act
on.
Not
used
by
this
drive

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
iron
comm
sta

reboot(*node*)

Per-
form
a
hard
re-
boot
of
the
task
node

Parameter

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing

the
node
to
act
on.

- **time**
Time
to
wait
for
the
node
to
be-
com-
pow-
ered
on.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu-
on
mal-
form-
pa-
ram-
e-
ter(s)

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-
e-
ter

is
miss
ing.

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

set_power

Set
the
pow
state
of
the
task.
node

Parameter

- **task**
A
Task
ager

in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **power**
Any
power
state
from
ironic
compute
state

- **time**
Tim
to
wait
for
the
node
to
reac
the
re-
ques
state

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CER
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

validate

Val-

i-
date
the
driv
in-
for
ma-
tion
need
by
the
iBM
driv

Param

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-

ironic.drivers.modules.ibm.raid module

Valu
on
miss
ing
pa-
ram-
e-
ter(s)

iBM
RAI
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

class ir

Base
ironic
drivers
base
RAI

Im-
ple-
men-
ta-
tion
of
RAI
In-
ter-
face
for
iBM

```
RAID_API
{'descr:
volumes
'True'.
"Setting
specifi
False},
indicat
configu
'raid_co
'require
```

apply_co

Ap-
plies
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

Paramete

- **task**
A
Task
ager
in-
stan
- **rai**
The
RAID
con-
fig-
u-
ra-
tion
to
ap-
ply.
- **cre**

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate

creating the new configuration.

to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

plete.

create_c

Cre-
ate
a
RAI
con-
fig-
u-
ra-
tion.
This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
a
Task
ager
in-
stan
- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing

erwise, no root volume is created. Default is True.

ated. Default is True.

creating the new configuration. Default is False.

RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

volume and/or non-root volumes.

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or
emp
af-
ter
skip
ping
root

Raises
IBM
CER
ror,
on
fail-
ure
to
ex-
e-
cute
step

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame
task
a
Task
ager
in-
stan

con-
tain-
ing
the
node
to
act
on.

Returns

state
if
clear
ing
op-
er-
a-
tion
in
prog
asyn
chro
or
state
if
de-
ploy
op-
er-

ation in progress synchronously or None if it is completed.

Raises

IBM
CEr
ror,
on
fail-
ure
to
ex-
e-
cute
step

get_prop

Re-
turn
the
prop
er-
ties
of

the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

ironic.drivers.modules.ibmutils module

iBM
Driv
com
mon
utils

ironic.d

Dec
o-
ra-
tor
to
han-
dle
iBM
clie
ex-
cep-
tion.

Dec
o-
rate
func
tion:
mus
take
a
Tas

as
the
first
pa-
ram-
e-
ter.

ironic.d

Pars
the
in-
for-
ma-
tion
re-
quir
for
Iron
to
con-
nect
to
iBM

**Paramet
nod**

an
Iron
node
ob-
ject

Returns

dic-
tio-
nary
of
pa-
ram-
e-
ters

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on

mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.d

ironic.drivers.modules.ibmcc.vendor module

iBM
Ven-
dor
In-
ter-
face

class ir

Base
iro
dri
bas
Ven

boot_up.

List
boot
type
or-
der
of

the
node

Parameter

- **task**
A Task object representing a task in a task manager. The task is an instance of the Task class and contains information about the task, such as the name, the priority, and the status. The task is used to schedule and execute tasks.
- **kwargs**
Not used.

Raises

In-ValidParameterError
If the value of the `kwargs` parameter does not contain a method.

Raises

MissingParameterError
If the value of the `kwargs` parameter is not a dictionary.

Raises

IBM

C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns

A
dic-
tio-
nary
con-
tain-
ing
node
boot
up
se-
quer
in
as-
cend
ing
or-
der.

get_prop

Re-
turn

the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_rai

List
RAI
con-
troll
sum
mar
info
of
the
node

Parame

- **task**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

- **kwargs**
Not
used

Raises

IBM
C-
Con-
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns

A
list
of
dic-
tio-
nar-
ies,
ev-
ery
dic-
tio-
nary
rep-

summary of node.

re-
sent
a
RAI
con-
troll

validate

Val-
i-
date
vend
spec
ac-
tions

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **task**
A
task
from
Task
ager
- **met**
Met
to
be
val-
i-
date
-

faces.

kwargs
Info
for
ac-
tion.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwargs
does
not
con-
tain
meth

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu

Module contents

`ironic.drivers.modules.ilo` package

Submodules

`ironic.drivers.modules.ilo.bios` module

iLO
BIO
In-
ter-
face

class `ir`

Base
iro
dri
bas
BIO

apply_c

Ap-
plies
the
pro-
vide
con-
fig-
u-
ra-
tion
on
the
node

Paramet

- **task**
a
Task
ager
in-
stan
- **set**

Set-
tings
in-
tend
to
be
ap-
plied
on
the
node

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

cache_b:

Stor
the
BIO
set-

tings
in
the
data

Parameter
task
a
Task
ager
in-
stan

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

factory.
Re-
set
the

BIO
set-
tings
to
fac-
tory
con-
fig-
u-
ra-
tion.

Parame
task
a
Task
ager
in-
stan

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy

step

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

validate

Che
that
driv
con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
driv
prop
erty
of
the

dentials information.

sup-
plie
task
node
con-
tain
the
re-
quir
cre-

Parame
task
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-
e-
ters
are
not
valid

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-

ironic.drivers.modules.ilo.boot module

ram-
e-
ter
is
miss
ing.

Boo
In-
ter-
face
for
iLO
driv
and
its
sup-
port
ing
meth
ods.

class ir

Base
iro
dri
mod
pxe
PXE

clean_up

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
PXE
en-

unlinks the instance kernel/ramdisk in the nodes directory in tftproot and removes its PXE config. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

vi-
ron-
men-
that
was
setu
for
boot
ing
the
in-
stan
It

Parame

task
a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO
faile

prepare

Pre-
pare
the
boot
of
in-
stan
This

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parame
task
a
task
from
Task
ager

Returns
Non

Raises
Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO
faile

prepare_
Pre-

vant information from the nodes driver_info and instance_info.

pare
the
boot
of
Iron
ram
us-
ing
PXE
This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

Parame

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass

or instance_info.

to
the
ram

Returns
Non

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
driv

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises
Iron
icEx
cep-

the node.

tion.
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO
faile

class ir

Base
iro
dri
bas
Boo

capabil:

clean_up

Clea
up
the
boot
of
in-

stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

Parame

task
A
task
from
Task
ager

Returns

Non

clean_up

Clea
up
the
boot
of
iron
ram

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu

for
boot
ing
the
de-
ploy
ram

Parameters

task

A
task
from
Task
ager

Returns

Non

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

prepare

Pre-
pare
the
boot
of

tion from the nodes instance_info. It does the following depending on boot_option for deploy:

image, then it sets the node to boot from disk.

in-
stan-

This
meth
pre-
pare
the
boot
of
the
in-
stan-
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

- If the boot re-ques for this de-ploy is lo-cal or im-age is a who disk

- Oth-er-wise it find

sets the URL as the boot ISO to boot the instance image.

the
boot
ISO
sets
the
node
boot
op-
tion
to
UE-
FI-
HTT
and

Parameters

task
a
task
from
Task
ager

Returns

None

Raises

Ilo-
Op-
er-
a-
tion
if
some
op-
er-
a-
tion
on
iLO
failure

Raises

In-
stan-
ploy
Fail-
ure,
if
its
try
to

boot
iSCSI
vol-
ume
in
BIO
boot
mod

prepare_

Pre-
pare
the
boot
of
de-
ploy
ram
us-
ing
UEFI
HTT
boot

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

vant information from the nodes driver_info and instance_info.

Parame

- **task**
a

task
from
Task
ager

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns
Non

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
drive

or instance_info.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if

the node.

som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO
faile

validate

Val-
i-

the required information for this interface to function.

date
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

This
meth
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

Parameter

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parameter

task
A

Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-

eters

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
that
the
node
has
re-
quir

prop
er-
ties
for
res-
cue.

Parame

task

a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-

eters

class ir

Base
iro
dri
bas
Boo

capabil:

clean_up

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan
It
ejec

virtual media. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

Parame

task
a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-

er-
a-
tion
on
iLO
faile

clean_up

Clea
up
the
boot
of
iron
ram

This
meth
clea
up
vir-
tual
me-
dia
de-
vice
setu
for
the
de-
ploy
or
res-
cue
ram

Parame

task
a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tion

if
som
op-
er-
a-
tion
on
iLO
faile

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot

tion from the nodes instance_info. It does the following depending on boot_option for deploy:

get info and node to boot from UefiTarget boot device.

of
the
in-
stan-
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

- If the boot mode is uefi and its booting from volume then it sets the iSCSI tar-

- If not booting from volume and the boot requires for this

image is a whole disk image, then it sets the node to boot from disk.

ISO to the bare metal and then sets the node to boot from CDROM.

de-
ploy
is
lo-
cal
or

- Oth-
er-
wise
it
finds
the
boot
ISO
to
boot
the
in-
stan-
ce
im-
age,
at-
tach
the
boot

Parameter
task
a
task
from
Task
ager

Returns
Non

Raises
Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion

on
iLO
fail

Raises

In-
stan-
ploy
Fail-
ure,
if
its
try
to
boot
iSCS
vol-
ume
in
BIO
boot
mod

prepare_

Pre-
pare
the
boot
of
de-
ploy
rame
us-
ing
vir-
tual
me-
dia.

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-

vant information from the nodes driver_info and instance_info.

cue
rame
af-
ter
read
ing
rel-
e-

Parameters

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss

or instance_info.

the node.

ing
in
node
drive

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

Raises

Ilo-
Op-
er-
a-

tion)
if
som
op-
er-
a-
tion
on
iLO
faile

validat

Val-
i-
date
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

if
som
in-
for-
ma-
tion
is
in-
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
ker-
nel_
and
ram
are
miss
ing
in
the
Glan
im-

age or kernel and ramdisk not provided in instance_info for non-Glance image.

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

**Parameter
task**

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-

eters

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
that
the
node
has
re-

quir
prop
er-
ties
for
res
cue.

Parame

task
a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-

eters

class ir

Base
iro
dri
mod
ipx
iPX

clean_up

Clea
up
the
boot
of
in-
stan
This
meth
clea
up
the
PXE
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan
It

unlinks the instance kernel/ramdisk in the nodes directory in tftproot and removes its PXE config. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

Parame

task
a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-

er-
a-
tion
on
iLO
faile

prepare_

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

Parameter

task
a
task
from
Task
ager

Returns

Non

Raises

Ilo-

Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO
faile

prepare.

Pre-
pare
the
boot
of
Iron
ram
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

vant information from the nodes driver_info and instance_info.

Parame

-

or instance_info.

task
a
task
from
Task
ager

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns
Non

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
driv

Raises
In-
valic
Pa-
ram-
e-
ter-

the node.

Valu
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO
faile

ironic.d

information for this driver to deploy images to the node.

Gets
the
drive
spe-
cific
Node
de-
ploy
men
info.

This
meth
val-
i-
date
when
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-
quire

Parameter

- **node**
a
sin-
gle
Node
- **model**
La-
bel
in-
di-
cat-
ing
a

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

ing.

de-
ploy
or
res-
cue
op-
er-
a-
tion
be-
ing

Returns

A
dict
with
the
drive
val-
ues.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss

ironic.d

Com
mon
prep
tory
step
for

ables secure boot, if it is in enabled state. 3. Updates boot_mode capability to uefi if secure boot is requested. 4. Changes boot mode of the node if secure boot is disabled currently.

all
iLO
driv

This
meth
per-
form
com
mon
prep
tory
step
re-
quir
for
all
driv
1.
Pow
off
node
2.
Dis-

Paramet

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-

ironic.drivers.modules.ilo.common module

a-
tion
on
iLO
faile

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
iLO
mod
ules

ironic.d
Nod
is
in
Fin-
ishe
Post
post
state

ironic.d
'InPostD
Nod
is
in
In-
Post
Dis-
cov-
eryC
post
state

ironic.d
Nod
is

in
In-
Post
post
state

ironic.d
Nod
is
in
Null
post
state

ironic.d
Nod
is
in
Pow
post
state

ironic.d
Nod
is
in
Re-
set
post
state

ironic.d
Nod
is
in
Un-
know
post
state

ironic.d
'legacy
Nod
sup-
port
both
lega
BIO
and
UEF
boot
mod

ironic.d
'legacy'
Nod
sup-
port
only
lega
BIO
boot
mod

ironic.d
Nod
sup-
port
only
UEF
boot
mod

ironic.d
Add
cer-
tifi-
cate
to
the
node
Add
cer-
tifi-
cate
to
the
node
base
on
the
driv
info
pro-
vide

Paramet

- **task**
a
Task
ager
in-

configured in `webserver_verify_ca` will be added to the node.

stand-
con-
tain-
ing
the
node
to
act
on.

- **cer-**
List
of
cer-
tifi-
cate
to
be
adde
to
the
node
If
Non-
cer-
tifi-
cate
from
path

Raises
Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises
Ilo-
Op-
er-
a-

server.

valid.

tion)
Sup-
port
if
re-
triev
ing
post
state
is
not
sup-
port
on
the

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
in-

ironic.d

At-
tach
the
give
url
as
vir-
tual
me-
dia
on

the
node

Parameter

- **node**
an
iron
node
ob-
ject.
- **dev:**
the
vir-
tual
me-
dia
de-
vice
to
at-
tach
- **url**
the
http
url
to
at-
tach
as
the
vir-
tual
me-
dia
de-
vice

Raises
Ilo-
Op-
er-
a-
tionl
if
in-
sert
vir-

tual
me-
dia
faile

ironic.d

Clea
a
node
af-
ter
a
vir-
tual
me-
dia
boot

This
meth
clea
up
a
node
af-
ter
a
vir-
tual
me-
dia
boot
It
dele
the
flopp
im-
age

if it exists in CONF.ilo.swift_ilo_container or web server. It also ejects both virtual media cdrom and virtual media floppy.

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the

node
to
act
on.

ironic.d

Clea
any
cer-
tifi-
cate
add
to
the
node

Clea
the
cer-
tifi-
cate
add
to
the
node
as
part
of
any
Iron
op-
er-
a-
tion

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

present on the node will be removed.

on.

- **cert**
List
of
cer-
tifi-
cate
to
be
re-
mov
from
node
If
Non
all
the
cer-
tifi-
cate

Raises

Ilo-
Op-
er-
a-
tionl
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Ilo-
Op-
er-
a-
tionl
Sup-
port
if
re-
triev
ing
post

server.

state
is
not
sup-
port
on
the

ironic.d

Up-
load
the
give
im-
age
to
swif

This
meth
copi
the
give
im-
age
to
swif

Paramet

- **sou**
The
ab-
so-
lute
path
of
the
im-
age
file
whic
need
to
be
copi
to
swif
-

des
The
nam
of
the
ob-
ject
that
will
con-
tain
the
copi
im-
age.

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi
fails

Returns

temp
url
from
swif
af-
ter
the
sour
im-
age
is
up-
load

ironic.d

Cop
the
give
im-

to the image else the deploy fails as the image file at the web_server url is inaccessible.

server root.

age
to
the
http
web
serv

This
meth
meth
copi
the
give
im-
age
to
the
http
lo-
ca-
tion.
It
en-
able
read
writ
ac-
cess

Paramet

- **source**
The
ab-
so-
lute
path
of
the
im-
age
file
which
need
to
be
copi
to
the
web

-

des
The
nam
of
the
file
that
will
con-
tain
the
copi
im-
age.

Raises

Imag
U-
ploa
Fail
ex-
cep-
tion
if
copy
ing
the
sour
file
to
the
web
serv
fails

Returns

im-
age
url
af-
ter
the
sour
im-
age
is
up-
load

ironic.d

Re-

mov
the
tem-
po-
rary
flopp
im-
age.

It
re-
mov
the
flopp
im-
age
cre-
ated
for
de-
ploy
:par
node
an
iron
node
ob-
ject.

ironic.d

Ejec
vir-
tual
me-
dia
de-
vice

This
meth
eject
vir-
tual
me-
dia
flopp
and
cdro

Paramet
tas!
a

tual media floppy or cdrom.

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns
Non

Raises
Ilo-
Op-
er-
a-
tionl
if
som
er-
ror
was
en-
cour
tere
whil
try-
ing
to
ejec
vir-

ironic.d

Get
the
cur-
rent
boot
mod
for
a
node

Paramet
nod
an

iron
node
ob-
ject.

Raises

Ilo-
Op-
er-
a-
tionl
if
faile
to
fetch
boot
mod

Raises

Ilo-
Op-
er-
a-
tionl
Sup-
port
if
node
does
not
sup-
port
get-
ting
pend
ing
boot
mod

`ironic.d`

Gets
an
Ilo-
Clie
ob-
ject
from
pro-
liant
tils
li-
brar

tions on the iLO.

Give
an
iron
node
ob-
ject,
this
meth
give
back
a
Ilo-
Clie
ob-
ject
to
do
op-
er-
a-

Paramet

nod
an
iron
node
ob-
ject.

Returns

an
Ilo-
Clie
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
in-
puts

Raises

Mis:

the node

ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on

ironic.d

Re-
triev
cur-
rent
en-
able
state
of
UEF
se-
cure
boot
on
the
node

Re-
turn
the
cur-
rent
en-
able
state
of
UEF
se-
cure
boot
on

the
node

Parameter

task
a
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
iLO
pa-
ram-
e-
ter
is
miss
ing.

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Ilo-
Op-
er-
a-

tion)
Sup-
port
if
UEF
se-
cure
boot
is
not
sup-
port

Returns

Boo
valu
in-
di-
cat-
ing
cur-
rent
state
of
UEF
se-
cure
boot
on
the
node

ironic.d

Get
the
cur-
rent
state
of
sys-
tem
POS

Paramet

node
an
iron
node
ob-
ject.

Returns

POS
state
of
the
serv
The
val-
ida
state
are:-

null,
Un-
know
Re-
set,
Pow
In-
Post

InPostDiscoveryComplete and FinishedPost.

Raises

Ilo-
Op-
er-
a-
tionl
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Ilo-
Op-
er-
a-
tionl
Sup-
port
if
re-
triev
ing
post
state
is

server.

information for this driver.

not
sup-
port
on
the

ironic.d

Gets
the
drive
spe-
cific
Node
info.

This
meth
val-
i-
date
when
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-
quir

Paramet

node
an
iron
Node
ob-
ject.

Returns

a
dict
con-
tain-
ing

values).

in-
for-
ma-
tion
from
driv
(or
whe
ap-
pli-
ca-
ble,
con-
fig

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
pa-
ram-
e-
ters
are
in-
cor-
rect

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is

the node

in CONF.ilo.swift_ilo_container

miss
ing
on

ironic.d

Re-
mov
the
give
im-
age
from
swif

This
meth
re-
mov
the
give
im-
age
nam
from
swif

It
dele
the
im-
age
if
it
ex-
ists

Paramet

- **obj**
The
nam
of
the
ob-
ject
whic
need
to
be

re-
mov
from
swif

- **ass**
strin
to
de-
pict
the
com
po-
nent
this
ob-
ject
is
as-
so-
ci-
ated
to.

ironic.d

Re-
mov
the
give
im-
age
from
the
con-
fig-
ured
web
serv

This
meth
re-
mov
the
give
im-
age
from
the
http
lo-

ca-
tion.
if
the
im-
age
ex-
ists.

Paramet

obj

The
nam
of
the
im-
age
file
whic
need
to
be
re-
mov
from
the
web
serv
root

ironic.d

Re-
mov
(dele
the
file
or
list
of
files

This
meth
only
ac-
cept
sin-
gle
or
list
of
files

method removes (deletes) the file. If list of files is passed, this method removes (deletes) each of the files iteratively.

to
dele
If
sin-
gle
file
is
pass
this

Paramet

fil

a
sin-
gle
or
a
list
of
file
path

ironic.d

Sets
the
node
to
boot
us-
ing
boot
for
the
next
boot

Paramet

- **node**
an
iron
node
ob-
ject.

- **boot**
Nex
boot

mod
Raises
Ilo-
Op-
er-
a-
tionl
if
set-
ting
boot
mod
faile

ironic.d

En-
able
or
dis-
able
UEFI
Se-
cure
Boo
for
the
next
boot

En-
able
or
dis-
able
UEFI
Se-
cure
Boo
for
the
next
boot

Paramet

- **task**
a
task
from
Task

ager

- **fla**
Boo
valu
True
if
the
se-
cure
boot
to
be
en-
able
in
next
boot

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Ilo-
Op-
er-
a-
tion
Sup-
port
if
UEF
se-
cure
boot
is
not
sup-
port

ironic.d

Sets
up
sys-
tem
to
boot
from
UE-
FI-
HTI
boot
de-
vice

Sets
the
one-
time
boot
de-
vice
to
UE-
FI-
HTI
base
on
the
ar-
gu-
men-
sup-
plied

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

- **iso**
ISO
URI
to
be
set
to
boot
from

- **per-**
In-
di-
cate
when
the
sys-
tem
shou
be
set
to
boot
from
the
give
de-
vice
one-

time or each time.

Raises

Ilo-
Op-
er-
a-
tion)
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Ilo-

server.

Op-
er-
a-
tion
Sup-
port
if
re-
triev
ing
post
state
is
not
sup-
port
on
the

`ironic.d`

At-
tach
vir-
tual
me-
dia
and
sets
it
as
boot
de-
vice

This
meth
at-
tach
the
give
boot
ISO
as
vir-
tual
me-
dia,
pre-
pare
the

ramdisk in virtual media floppy.

ar-
gu-
men
for

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **iso**
a
boot
ISO
im-
age
href
to
at-
tach
to.
Sho
be
ei-
ther
of
be-
low:

-
A
Swi
ob-
ject
-

It

image object is present in CONF.ilo.swift_ilo_container;

shou
be
of
for-
mat
swi.
It
is
as-
sum
that
the

-
A
Glan
im-
age
-

It
shou
be
for-
mat
glan
/
<gl.
or
just
<gl.

-
An
HTT
URI

•
ram
the
op-
tions
to
be
pass
to
the
ram
in
vir-
tual
me-

dia
flopp

Raises

Im-
age-
Cre-
ation
Failure
if
it
failure
while
cre-
at-
ing
the
flopp
im-
age.

Raises

Illo-
Op-
er-
a-
tion
if
some
op-
er-
a-
tion
on
iLO
failure

ironic.d

Sets
up
the
node
to
boot
from
the
give
ISO
im-
age.
This

it via virtual floppy image.

meth
at-
tach
the
give
boot
on
the
node
and
pass
the
re-
quir
pa-
ram-
e-
ters
to

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **boot**
a
boot
ISO
im-
age
to
at-
tach
to.
Sho
be

ei-
ther
of
be-
low:

-
A
Swi
ob-
ject
-

It
shou
be
of
for-
mat
swi.
It
is
as-
sum
that
the

image object is present in CONF.ilo.swift_ilo_container;

-
A
Glar
im-
age
-

It
shou
be
for-
mat
glar
/
<gl.
or
just
<gl.

-
An
HTT
URI

•

nary. This is optional.

par
the
pa-
ram-
e-
ters
to
pass
in
the
vir-
tual
flopp
im-
age
in
a
dic-
tio-

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the
flopp
im-
age.

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with

Swi
fails

Raises

Ilo-
Op-
er-
a-
tion
if
at-
tach
ing
vir-
tual
me-
dia
faile

`ironic.d`

Up-
date
in-
stan
with
boot
mod
to
be
used
for
de-
ploy

This
meth
up-
date
in-
stan
with
boot
mod
to
be
used
for
de-
ploy
if
node
prop

do not have `boot_mode`. It sets the boot mode on the node.

er-
ties[

Parameter

task
Task
ob-
ject.

Raises

Ilo-
Op-
er-
a-
tion
if
set-
ting
boot
mod
faile

`ironic.d`

Up-
date
ipmi
prop
er-
ties
to
node
driv

Parameter

task
a
task
from
Task
ager

`ironic.d`

Val-
i-
date
se-
cu-
rity
pa-
ram-
e-

ter
with
valid
val-
ues.

Parameter

sec-
dict
ob-
ject
con-
tain-
ing
the
se-
cu-
rity
pa-
ram-
e-
ter
info

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
for
miss-
ing
field
(or
val-
ues)
in
se-
cu-
rity
pa-

parameter info.

Raises

In-
valid
Pa-
ram-
e-
ter-

Valu
for
un-
sup-
port
se-
cu-
rity
pa-
ram-
e-
ter

Returns

tu-
ple
of
se-
cu-
rity
para
ig-
nore
and
en-
able
pa-
ram-
e-
ters.

ironic.d

Ver-
i-
fies
chec
sum
(md
of
im-
age
file
agai
the
ex-
pect
one.

This
meth
gen-
er-

ates
the
chec
sum
of
the
im-
age
file
on
the
fly
and
ver-
i-
fies

it against the expected checksum provided as argument.

Paramet

- **ima**
lo-
ca-
tion
of
im-
age
file
who
chec
sum
is
ver-
i-
fied.
- **exp**
chec
sum
to
be
chec
agai

Raises
Im-
ageE
f-
Val-
i-

fails.

`ironic.drivers.modules.ilo.console` module

da-
tion-
Failo
if
in-
valid
file
path
or
ver-
i-
fi-
ca-
tion

iLO
De-
ploy
Driv
and
sup-
port
ing
meth
ods.

class `ironic.drivers.modules.ilo.console.ILOConsole`

Base
ironic.drivers.modules.ilo.console.ILOConsole

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
shel
linat

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

validate

Val-
i-
date
the
Nod
con-
sole
info.

Paramet

task
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-

e-
ter-
Valu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
whe
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

ironic.drivers.modules.ilo.firmware_processor module

Firm
file
pro-
ces-
sor

class ir

Base
obj

Firm
im-
age
lo-
ca-
tion
class

This
class
acts
as
a
wrap

helps in removing the firmware files from their respective locations, made available for firmware update operation.

location it wraps.

per
class
for
the
firm
im-
age
lo-
ca-
tion.
It
pri-
mar-
ily

remove()

Ex-
pose
meth
to
re-
mov
the
wrap
firm
file

This
meth
gets
over
rid-
den
by
the
re-
mov
meth
for
the
re-
spec
tive
type
of
firm
file

class ir

in compact format) and makes it ready for firmware update operation. In future, methods can be added as and when required to extend functionality for different firmware file types.

Base
obj

Firm
file
pro-
ces-
sor

This
class
help
in
dow
load
ing
the
firm
file
from
url,
ex-
tract
ing
the
firm
file
(if
its

process

Pro-
cess
the
firm
file
from
the
url

This
is
the
tem-
plate
meth
which
dow
load
the

extracts the firmware and makes it ready for firmware update operation. `_download_fw_to` method is set in the firmware processor object creation factory method, `get_fw_processor()`, based on the url type. `:param node:` a single Node. `:param expected_checksum:` checksum to be checked against. `:returns:` wrapper object of raw firmware image location `:raises:` `IloOperationError`, on failure to process firmware file. `:raises:` `ImageDownloadFailed`, on failure to download the original file. `:raises:` `ImageRefValidationFailed`, on failure to verify the checksum. `:raises:` `SwiftOperationError`, if upload to Swift fails. `:raises:` `ImageUploadFailed`, if upload to web server fails.

firm
file
from
url,
ver-
i-
fies
chec
sum
and

ironic.d

Val-
i-
date
the
firm
im-
age
info
and
re-
turn
the
re-
triev
val-
ues.

Paramet

firm
dict
ob-
ject
con-
tain-
ing
the
firm
im-
age
info

Raises
Miss

ing-
Pa-
ram-
e-
ter-
Valu
for
miss
ing
field
(or
val-
ues)
in
im-
age
info.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
for
un-
sup-
port
firm
com
po-
nent

Returns

tu-
ple
of
firm
url,
chec
sum
com
po-
nent
whe
the
firm
up-
date
is

ing url as `swift://containername/objectname`.

ilo
base
ironic.d
Gets
swift
temp
url.
It
gen-
er-
ates
a
temp
url
for
the
swift
base
firm
url
to
the
tar-
get
file.
Ex-
pect

Parameter
par:
Pars
url
ob-
ject.

Raises
Swi
Op-
er-
a-
tion
on
fail-
ure
to
get
url
from
swif

ironic.d

Ver-
i-
fies
the
firm
up-
date
ar-
gu-
men

ironic.drivers.modules.ilo.inspect module

iLO
In-
spec
In-
ter-
face

class ir

Base
iro
dri
bas
Ins

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty

nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware
to
get
the
hard
ware
prop
er-
ties.

In-
spec
hard
ware
to
get
the
es-
sen-
tial
and
ad-
di-
tiona
hard
ware
prop
er-
ties.
It

fails if any of the essential properties are not received from the node. It doesnt fail if node fails to return any capabilities as the capabilities differ from hardware to hardware mostly.

Paramet
task
a
Task
ager
in-
stan

cessfully.

Raises

Har
ware
spec
tion-
Fail-
ure
if
es-
sen-
tial
prop
er-
ties
coul
not
be
re-
triev
suc-

Raises

Ilo-
Op-
er-
a-
tion
if
sys-
tem
fails
to
get
pow
state

Returns

The
re-
sult-
ing
state
of
in-
spec
tion.

validate

Che
that
driv

dentials information.

con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
driv
prop
erty
of
the
sup-
plie
task
node
con-
tains
the
re-
quir
cre-

Parame
task
a
task
from
Task
ager

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-

e-
ters
are
not
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.drivers.modules.ilo.management module

iLO
Man
age-
men
In-
ter-
face

class ir

Base
iro
dri
mod
ilo
man
Ilo

erase_d

Eras
all

drives in the node. This erase cannot be performed on logical drives.

the
driv
on
the
node
This
meth
per-
form
out-
of-
band
san-
i-
tize
disk
eras
on
all
the
sup-
port
phys
i-
cal

Parame

task
a
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
ar-
gu-
men
are
in-

valid
Raises
Ilo-
Er-
ror
on
an
er-
ror
from
iLO

one_button
Eras
the
who
sys-
tem
se-
cure
The
One
butte
se-
cure
eras
pro-
cess
re-
sets
iLO
and
dele
all
li-
cens
store
there
re-
sets

BIOS settings, and deletes all Active Health System (AHS) and warranty data stored on the system. It also erases supported non-volatile storage data and deletes any deployment setting profiles.

Parameter
task
a
Task
ager
in-
stan

Raises

Ilo-
Er-
ror
on
an
er-
ror
from
iLO

class ir

Base
iro
dri
bas
Man

activate

Ac-
ti-
vate
iLO
Ad-
vanc
li-
cens

Parame

task
a
Task
ager
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
ar-
gu-
men

are
in-
valid

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

clear_is

Un-
set
iSCS
de-
tails
of
the
sys-
tem
in
UEFI
boot
mod

Parame

task
a
task
from
Task
ager

Raises

Ilo-
Com
man
Not-
Sup-
port
e-

d-
In-
Bios
ror
if
sys-
tem
in
BIO
boot
mod

Raises

Ilo-
Er-
ror
on
an
er-
ror
from
iLO

clear_se

Clea
all
se-
cure
boot
keys

Clea
all
the
se-
cure
boot
keys
This
op-
er-
a-
tion
is
sup-
port
only
on
HP
Pro-
liant

Gen9 and above servers.

Parameter
task
a
task
from
Task
ager

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

flash_f
De-
ploy
step
to
Up-
date
the
firm

us-
ing
Sma
Up-
date
Man
ager
(SU

Parame

task
a
Task
ager
ob-
ject.

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

get_boot

Get
the
cur-

rent
boot
de-
vice
for
a
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Parameter

task
a
task
from
Task
ager

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
iLO
pa-
ram-
e-
ter
is
miss-
ing.

Raises

Ilo-
Op-
er-

a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
ironic
comm
boo
or
Non
if

it is unknown.

persist

Whe
the
boot
de-
vice
will
per-

unknown.

sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_boot

Get
the
cur-
rent
boot
mod
for
a
node

Pro-
vide
the
cur-
rent
boot
mod
of
the
node

Parameter

task
A
task
from
Task
ager

Raises

Ilo-
Op-
er-
a-
tion
on
an

er-
ror
from
Ilo-
Clie
li-
brar

Returns

The
boot
mod
one
of
iron
comm
boo
or
Non
if
it
is
un-
know

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_secu

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

Paramete

task
A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from

Ilo-
Clie
li-
brar

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
hard
ware

Returns

Boo

get_sens

Get
sen-
sors
data

Paramete

task

a
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting

the
sen-
sor
data
fails

Raises

Fail
ToP
eSer
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-

ram-
e-
ter
is
miss
ing.

Returns

re-
turn
a
dict
of
sen-
sor
data
group
by
sen-
sor
type

get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Paramete

task
a
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice

de-
fine
in
ironic
command
book

get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter

task
a
task
from
Task
manager

Raises

Ilo-
Op-
er-
a-
tion
if
any
ex-
cep-
tion
hap-
pens
in
pro-
liant
tils

Returns

A
list
with
the
sup-
port

boot
de-
vice
de-
fined
in
iro
comm
boo

inject_r

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

tion.

on.

Raises

Ilo-
Com
man
Not-
Sup-
port
ed-
Er-
ror
if
sys-
tem
does
not
sup-
port
NM
in-
jec-

Raises

Ilo-
Er-
ror
on
an
er-
ror
from
iLO

Returns

Non

reset_b

Re-
sets
the
BIO
set-
tings
to
de-
fault
val-
ues.
Re-
sets

HP Proliant Gen9 and above servers.

BIO
to
de-
fault
set-
tings
This
op-
er-
a-
tion
is
cur-
rently
sup-
port
only
on

Parameter
task
a
task
from
Task
ager

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
In-
stan
ploy
Fail-
ure,
on
fail-

ure
to
ex-
e-
cute
of
de-
ploy
step

reset_i

Re-
sets
the
iLO

Parame

task
a
task
from
Task
ager

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-

e-
cute
of
de-
ploy
step

reset_i

Re-
sets
the
iLO
pass
wor

Parame

- **task**
a
task
from
Task
ager

- **char**
Valu
for
pass
wor
to
up-
date
on
iLO

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea

step

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

reset_s

Re-
set
se-
cure
boot
keys
to
man
u-
fac-
tur-
ing
de-
fault

Re-
sets
the
se-
cure
boot
keys
to
man
u-
fac-
tur-
ing
de-
fault
This

is supported only on HP Proliant Gen9 and above servers.

op-
er-
a-
tion

Parame
task
a
task
from
Task
ager

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

security
Up-
date

the
se-
cu-
rity
pa-
ram-
e-
ters.

Parame

task

a
Task
ager
ob-
ject.

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

-

task
a
task
from
Task
ager

not. Default: False.

- **dev:**
the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
[ironic](#)
[command](#)
[book](#)

- **per:**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
an
in-

valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

set_boot

Set
the
boot
mod
for
a

node
Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Parameter

- **task**
A
task
from
Task
ager
- **mod**
The
boot
mod
one
of
iron
com
boo

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod
is
spec

i-
fied.

Raises

Ilo-
Op-
er-
a-
tion!
if
set-
ting
boot
mod
faile

set_iscs

Set
iSCS
de-
tails
of
the
sys-
tem
in
UEFI
boot
mod

The
ini-
tia-
tor
is
set
with
the
tar-
get
de-
tails
like
IQN
LUN
IP,
Port
etc.
:par:
task

a task from TaskManager. :raises: MissingParameterValue if a required parameter is missing. :raises: IloCommandNotSupportedInBiosError if system in BIOS boot mode. :raises: IloError on an error from

iLO.

set_secu

Set the current secure boot state for the node

Parame

- **task**
A task from Task Manager

- **state**
A new state as a bool

Raises

Missing-Parameter-Error
ValueError
if a required parameter is missing

ing

Raises

Ilo-

Op-

er-

a-

tionl

on

an

er-

ror

from

Ilo-

Clie

li-

bran

Raises

Un-

sup-

port

ed-

Driv

ten-

sion

if

se-

cure

boot

is

not

sup-

port

by

the

hard

ware

update_2

Up-

date

the

Autl

Fail-

ure

Log

ging

Thre

old

se-

cu-

rity
pa-
ram-
e-
ter.

Parame

tas

a
Task
ager
ob-
ject.

update_:

Up-
date
the
firm

Parame

tas

a
Task
ager
ob-
ject.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
up-
date
firm
mod
is
not
ilo.
Ever
ap-
pli-
ca-

ble for invalid input cases.

Raises

Nod
Clea

ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clear
step

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

update_:

Clea
step
to
up-
date
the
firm
us-
ing
Sma
Up-
date
Man
ager
(SU

Paramet

task
a

Task
ager
ob-
ject.

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

update_r

Up-
date
the
Min
i-
mun
Pass
wor
Leng
se-
cu-
rity
pa-
ram-
e-
ter.

dentials information.

Parameter
task
a
Task
ager
ob-
ject.

validate
Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
task
node
con-
tains
the
re-
quir
cre-

Parameter
task
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-
e-
ters
are
not
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.drivers.modules.ilo.power module

iLO
Pow
Driv

class ir

Base
iro
dri
bas

Pow

get_pow

Gets
the
cur-
rent
pow
state

Parameter

- **task**
a
Task
ager
in-
stan
- **node**
The
Nod

Returns

one
of
ironic
comm
sta
POV
POV
or
ER-
ROF

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
cre-
den-

tials
are
miss
ing.

Raises

Ilo-
Op-
er-
a-
tionl
on
an
er-
ror
from
Ilo-
Clie
li-
brar

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list

of
the
sup-
port
pow
state

Parame

task

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
cur-
rentl
not
used

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
ironic
comm
sta

reboot (a

Re-
boot
the
node

Parame

•

task

a

Task

ager

in-

stand

•

time

time

out

(in

sec-

onds

Un-

sup-

port

by

this

in-

ter-

face

Raises

Pow

er-

State

Fail-

ure

if

the

fi-

nal

state

of

the

node

is

not

POV

Raises

Ilo-

Op-

er-

a-

tion

on

an

er-

ror

from

Ilo-

Client
library

set_power

Turns
the
current
power
state
on
or
off.

Parameters

- **task**
a Task object representing the instance
- **power**
The desired power state (POW or RE-BOO) from [ironic.common.constants](#)
- **timeout**
time out (in seconds). Un-supported by

this
in-
ter-
face

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Pow
er-
State
Fail-
ure
if
the
pow
coul
be
set
to

pow

validate

Che
if
node
con-
tains
the
re-
quir
iLO
cre-
den-
tials

Parame

- **task**
a
Task
ager
in-
stan

- **node**
Sin-
gle
node
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
cre-
den-
tials
are
miss
ing.

ironic.drivers.modules.ilo.raid module

iLO
RAID
spe-
cific
meth
ods

class `ir`

Base
ironic
driver
base
RAID

Im-
ple-
men-
ta-
tion
of
OOB
RAID
In-
ter-
face
for
iLO.

`apply_co`

Ap-
plies
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

Parameter

- `task`
A

ified in `raid_config`. Default value is True.

Task
ager
in-
stan

- **raid**
The
RAI
con-
fig-
u-
ra-
tion
to
ap-
ply.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns
state

plete.

if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_c

Cre-
ate
a
RAI
con-
fig-
u-
ra-
tion
on
a
bare
meta
us-
ing
ager
rame

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on

the
give
node

Paramete

- **task**
a
Task
ager
in-
stand
- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

erwise, no root volume is created. Default is True.

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root

ated. Default is True.

ter skipping root volume and/or non-root volumes.

vol-
ume
are
cre-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or
was
foun
to
be
emp
af-

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
clea
step

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure

to
ex-
e-
cute
de-
ploy
step

delete_c

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

task

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
clea
step

Raises

In-

stand
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
de-
ploy
step

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

ironic.drivers.modules.ilo.vendor module

Ven-
dor
In-
ter-
face
for
iLO
driv
and
its
sup-
port
ing
meth
ods.

class ir

Base
iro

dri
bas
Ven

Ven
spec
in-
ter-
face
for
iLO
de-
ploy
drive

boot_int

At-
tach
an
ISO
im-
age
in
glan
and
re-
boot
bare
meta

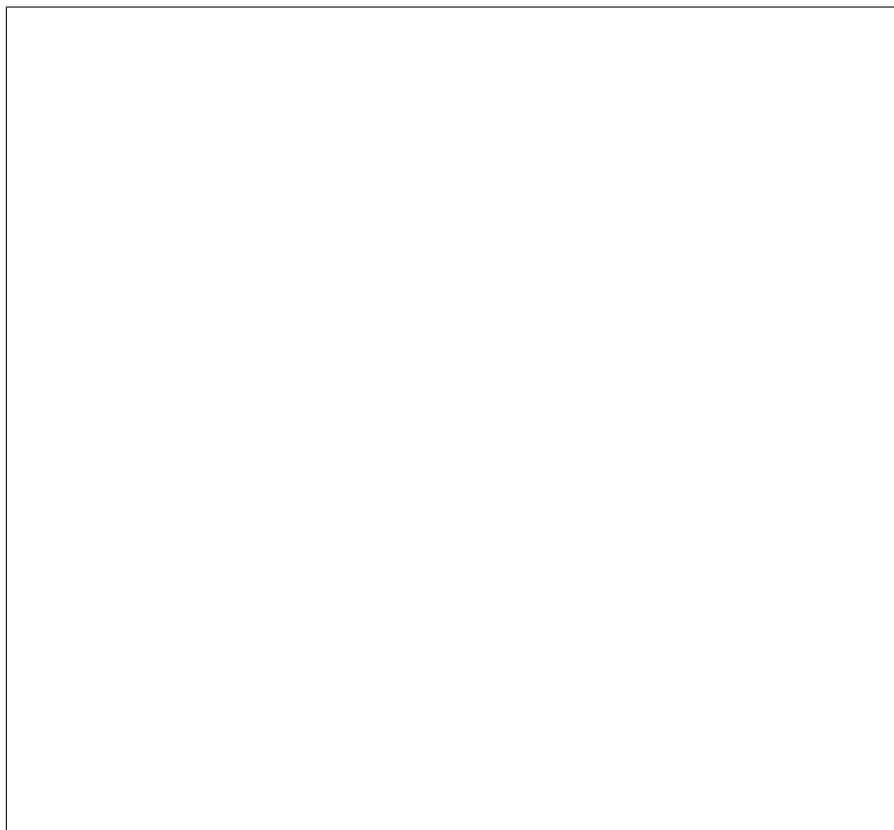
This
meth
ac-
cept
an
ISO
im-
age
href
(a
Glan
UUI
or
an
HTT
URI
at-
tach
it
as

virtual media and then reboots the node. This is useful for debugging purposes. This can be invoked

only when the node is in manage state.

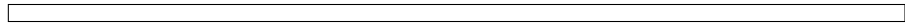
Parameters

- **task**
A Task object.
- **kwargs**
The arguments sent with vendor pass. The expected kwargs are:



(continues on next page)

(continued from previous page)



get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validate

Val-
i-
date
vend
spec
ac-
tions

Che
if
a
valid
ven-
dor
pass
meth
was
pass
and
val-
i-

the vendor passthru method.

date
the
pa-
ram-
e-
ters
for

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **met**
meth
to
be
val-
i-
date
- **kw**
kwa
con-
tain-
ing
the
ven-
dor
pass
meth
ods
pa-
ram-
e-
ters.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-
e-
ters
were
not
pass

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

Module contents

`ironic.drivers.modules.intel_ipmi` package

Submodules

ironic.drivers.modules.intel_ipmi.management module

In-
tel
IPM
Har
ware

Sup-
port
In-
tel
Spee
Se-
lect
Per-
for-
man
Pro-
file.

class ir

Base
iro
dri
mod
ipm
IPM

configur

Module contents

ironic.drivers.modules.irmc package

Submodules

ironic.drivers.modules.irmc.bios module

iRM
BIO
con-
fig-
u-
ra-
tion

spe-
cific
meth
ods

class `ir`

Base
iro
dri
bas
BIO

apply_co

Ap-
plies
BIO
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
take
the
BIO
set-
tings
from
the
set-
tings
para
and
ap-
plies
BIO
con-
fig-
u-
ra-

tion on the given node. After the BIOS configuration is done, `self.cache_bios_settings()` may be called to sync the nodes BIOS-related information with the BIOS configuration applied on the node. It will also validate the given settings before applying any settings and manage failures when setting an invalid

BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties.

dictionary as well.

Parameters

- **task**
a Task manager instance
- **set**
Dictionary containing the BIOS configuration. It may be an empty

Raises

IRM
C-
Op-
er-
a-
tion
ap-
ply
bios
set-
tings
failure

cache_bios

Storage
or

up-
date
BIO
set-
tings
on
the
give
node
This
meth
store
BIO
prop
er-
ties
to
the
bios
set-
tings
db

Parameter
task
a
Task
ager
in-
stan

Raises
IRM
C-
Op-
er-
a-
tion
get
bios
set-
tings
faile

Returns
Non
if
it
is
com
plete

factory_

Re-
set
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node

Parameter
task
a
Task
ager
in-
stan

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
BIO
re-
set.

get_prop

Re-
turn
the
prop
er-

ties
of
the
in-
ter-
face

validate

Val-
i-
date
the
drive
spec
Node
info.

This
meth
val-
i-
date
when
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-
quir

information for this driver to manage the BIOS settings of the node.

Parameter

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

on the node.

the driver_info property.

act
on.
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing
in

ironic.drivers.modules.irmc.boot module

iRM
Boo
Driv

class ir

Base
iro
dri
mod
pxe
PXE

iRM
PXE
boot

prepare_

Pre-
pare
the
boot
of
Iron
rame
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
ker-
nel/
af-
ter
read
ing
rel-
e-
vant

formation from the nodes `driver_info` and `instance_info`.

ters as kernel command-line arguments.

in-

Parame

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
pxe
driv
pass
thes
pa-
ram-
e-

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is

or instance_info.

node.

miss
ing
in
node
drive

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
de-
vice
op-
er-
a-
tion
faile
on
the

class ir

Base
iron

dri
bas
Boo
iro
dri
mod
irm
boo
IRM
iRM
Vir-
tual
Me-
dia
boot
relat
ac-
tions

capabil:

clean_up

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

Parame

task
a
task
from
Task
ager

Returns
Non

Raises
IRM
C-
Op-
er-
a-
tionl
if
iRM
op-
er-
a-
tion
faile

clean_up

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy

cue ramdisk.

or
res-

Parameters
task
a
task
from
Task
ager

Returns
Non

Raises
IRM
C-
Op-
er-
a-
tionl
if
iRM
op-
er-
a-
tion
faile

get_prop
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip

tion from the nodes database.

tion:
en-
tries

prepare_

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Paramete

task
a
task
from
Task
ager

Returns

Non

prepare_

Pre-
pare
the
de-
ploy

tual media cdrom.

or
res-
cue
ram
us-
ing
vir-
tual
me-
dia.
Pre-
pare
the
op-
tions
for
the
de-
ploy
or
res-
cue
ram
sets
the
node
to
boot
from
vir-

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
-

ram
the
op-
tion:
to
be
pass
to
the
ram

Raises

Im-
ageE
f-
Val-
i-
da-
tion-
Fail
if
no
im-
age
ser-
vice
can
han-
dle
spec
i-

fied href.

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the
flopp
im-
age.

Raises

In-

face or ManagementInterface fails.

valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion
of
the
Pow
er-
In-
ter-

Raises

IRM
C-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iRM
fails

validate

Val-
i-
date
the
de-
ploy
men
in-
for-
ma-
tion
for
the

task
node

Parameter

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
con-
fig
op-
tion
has
in-
valid
valu

Raises

IRM
C-
Shar
File
tem-
Not-
Mou
if
shar
file
sys-
tem
is
not
mou

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
in-
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
ker-
nel_
and
ram
are
miss
ing
in
the
Glan
im-

age, or if kernel and ramdisk are missing in the Non Glance image.

validate

Val-
i-
date
that
the
node
has
re-
quir
prop

er-
ties
for
res-
cue.

Parameters

task

a
Task
manager
instance
with
the
node
being
checked

Raises

Missing
parameter
value
if
node
is
missing
one
or
more
required
parameters

eters

Raises

Invalid
parameter
value
if
any
of

the
pa-
ram-
e-
ters
have
in-
valid
valu

class `ir`
Base
obj
Mix
in
class
for
vol-
ume
boot
con-
fig-
u-
ra-
tion
to
iRM

iRM
has
a
fea-
ture
to
set
up
re-
mote
boot
to
a
serv
This
fea-
ture
can
be
used

by VIOM (Virtual I/O Manager) library of SCCI client.

`ironic.d`

At-
tach
boot
ISO
for
a
de-
ploy
node
if
it
ex-
ists.

This
meth
check
the
in-
stan
info
of
the
bare
meta
node
for
a
boot
ISO
If
the
in-
stan

info has a value of key `irmc_boot_iso`, it indicates that `boot_option` is `netboot`. Therefore it attaches the boot ISO on the bare metal node and then sets the node to boot from virtual media `cdrom`.

Paramet

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

Raises

IRM
C-
Op-
er-
a-
tion
if
at-
tach
ing
vir-
tual
me-
dia
faile

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion
of
the
Man
age-
men
ter-

face fails.

ironic.d

Che
if
Shar
File
Sys-
tem
(NF
or
CIF
is

mou

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
con-
fig
op-
tion
has
in-
valid
valu

Raises

IRM
C-
Shar
File
tem-
Not-
Mou
if
shar
file
sys-
tem
is
not
mou

ironic.drivers.modules.irmc.common module

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-

ent
iRM
mod
ules

ironic.d

Gets
an
iRM
SCC
client

Give
an
iron
node
ob-
ject,
this
meth
give
back
a
iRM
SCC
client
to
do
op-
er-
a-
tions

on the iRMC.

Paramet

node
An
iron
node
ob-
ject.

Returns

sci.
par-
tial
func
tion
whic
take
a

SCC
com
man
para

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
in-
puts

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on

the node

ironic.d

Gets
iRM
SCC
re-
port

Give
an
iron
node

ob-
ject,
this
meth
give
back
a
iRM
SCC
re-
port

Parameter

node
An
iron
node
ob-
ject.

Returns

A
xml.
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
in-
puts

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-

the node.

for-
ma-
tion
is
miss
ing
on

Raises

sccl.
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

sccl.
if
SCC
faile

ironic.d

Get
the
cur-
rent
se-
cure
boot
mod

Paramet

node
An
iron
node
ob-
ject.

Raises

Un-
sup-
port
ed-
Driv
ten-

sion
if
se-
cure
boot
is
not
pres

Raises

IRM
C-
Op-
er-
a-
tionl
if
the
op-
er-
a-
tion
fails

ironic.d

Gets
the
spe-
cific
Node
drive
info.

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-

information for this driver.

quir

Paramet

node

An

iron

node

ob-

ject.

Returns

A

dict

con-

tain-

ing

in-

for-

ma-

tion

from

driv

and

de-

fault

val-

ues.

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

in-

valid

valu

is

con-

tain

in

the

driv

prop

erty.

Raises

Miss

ing-

erty.

Pa-
ram-
e-
ter-
Valu
if
som
man
tory
key
is
miss
ing
in
the
driv
prop

ironic.d

En-
able
or
dis-
able
UEF
Se-
cure
Boo

Paramet

- **node**
An
iron
node
ob-
ject.
- **enable**
Boo
valu
True
if
the
se-
cure
boot
to

be
en-
able

Raises

IRM
C-
Op-
er-
a-
tion
if
the
op-
er-
a-
tion
fails

`ironic.d`

Up-
date
ipmi
prop
er-
ties
to
node
driv

Paramet

task
A
task
from
Task
ager

`ironic.drivers.modules.irmc.inspect` module

iRM
In-
spec
In-
ter-
face

class ir

Base
iro

dri
bas
Ins
In-
ter-
face
for
out
of
band
in-
spec
tion.

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware

In-
spec
hard

ware
to
ob-
tain
the
es-
sen-
tial
hard
ware
prop
er-
ties
and
mac
ad-
dres

Parame

task

a
task
from
Task
ager

Raises

Har
ware
spec
tion-
Fail-
ure,
if
hard
ware
in-
spec
tion
faile

Returns

state
if
hard
ware
in-
spec
tion
suc-
ceed

validat

information for this driver.

Val-
i-
date
the
drive
spec
in-
spec
tion
in-
for-
ma-
tion.
This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Paramet
task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

on the node.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.d
<ironic_
SC2
sc2U
re-
turn
NIC
type

sc2UnitN
SYN

baseboard-controller(7) } ACCESS read-only STATUS mandatory DESCRIPTION Management node
class: primary: local operating system interface secondary: local management controller LAN interface
management-blade: management blade interface (in a blade server chassis) secondary-remote: remote
management controller (in an RSB concentrator environment) secondary-remote-backup: backup
remote management controller baseboard-controller: local baseboard management controller (BMC)
::= { sc2ManagementNodes 8 }

TAX
IN-
TE-
GEF
{
un-
know
pri-
mar
sec-
onda
man
blad
seco
remo
seco
remo
back

ironic.d
'1.3.6.1
SC2
sc2U
re-
turn
NIC
MA
ad-
dres

sc2UnitN
SYN
TAX
Phys
dres
AC-
CES
read
only
STA
TUS
man
tory
DE-
SCR
TIO

hardware (MAC) address ::= { sc2ManagementNodes 9 }

ironic.drivers.modules.irmc.management module

Man-
age-
men-
node

iRM-
Man-
age-
men-
Drive

class ir

Base
iro
dri
mod
ipm
IPM

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

Dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_secu

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

NOTE:

may
not
im-
ple-
men
that.

Parame

task
A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Driv
Op-

er-
a-
tion)
or
its
deri
tive
in
case
of
driv
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
driv
or

the hardware

Returns

Boo

get_sens

Get
sen-
sors
data
meth

It
gets
sen-
sor

dict format.

data
from
the
task
node
via
SCC
and
con-
vert
the
data
from
XML
to
the

Parameter

task
A
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
when
get-
ting
the
sen-
sor
data
fails

Raises

Fail
ToP
eSen
sor-
Data
when
pars
ing
sen-
sor

data
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

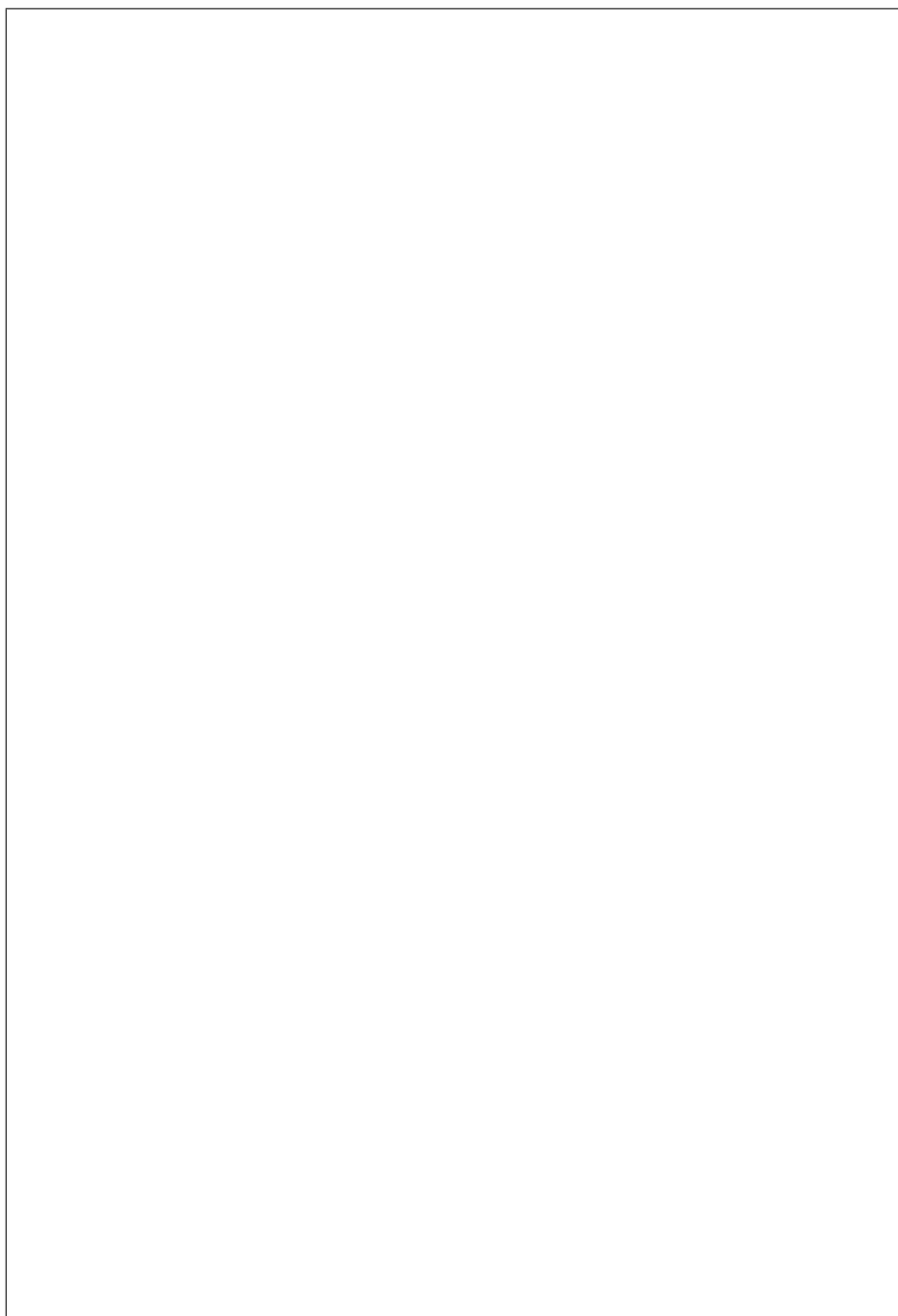
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

Re-
turn
a
con-
sis-
tent
for-
mat-

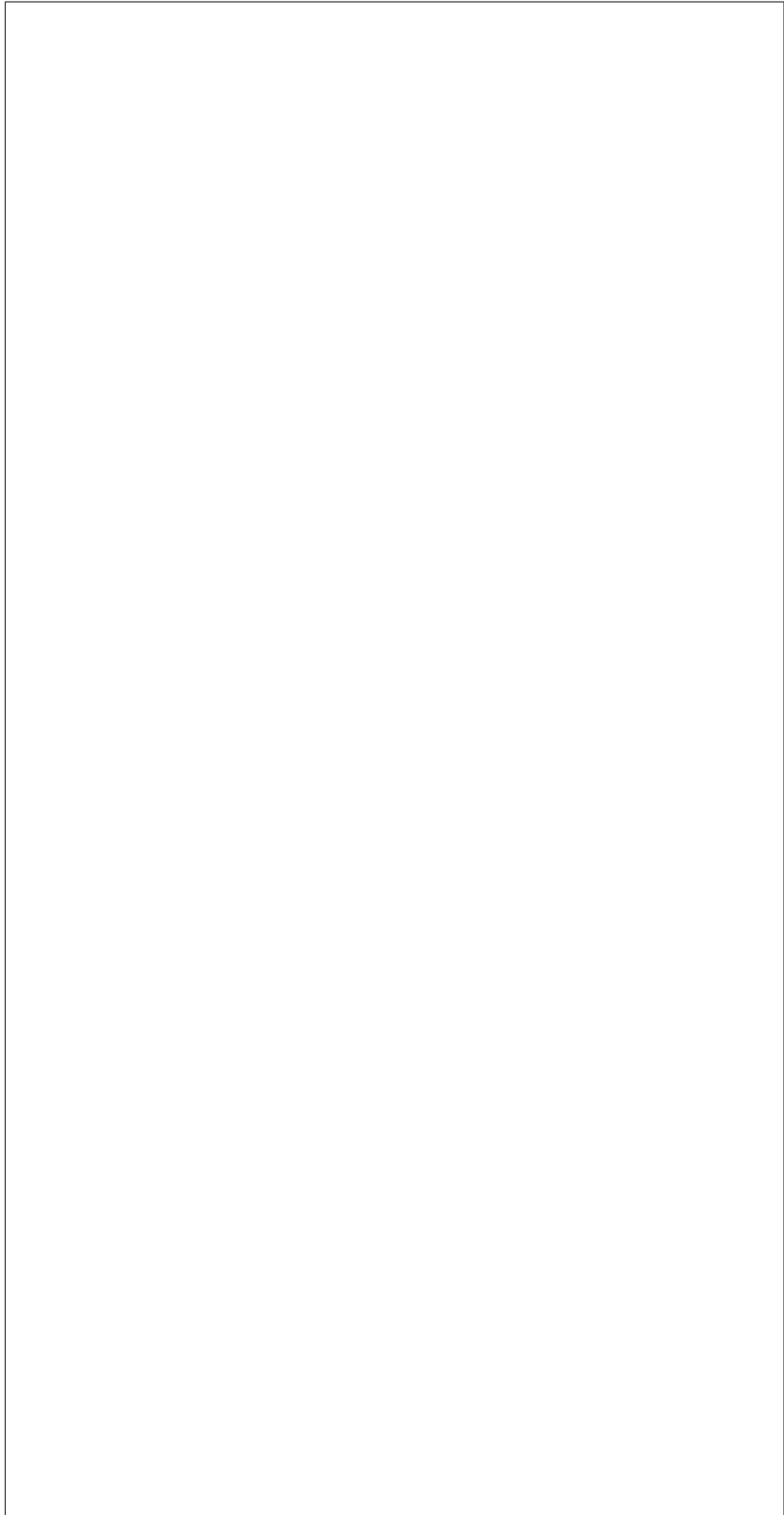
ted
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic

can be processed by Ceilometer. Example:



(continues on next page)

(continued from previous page)



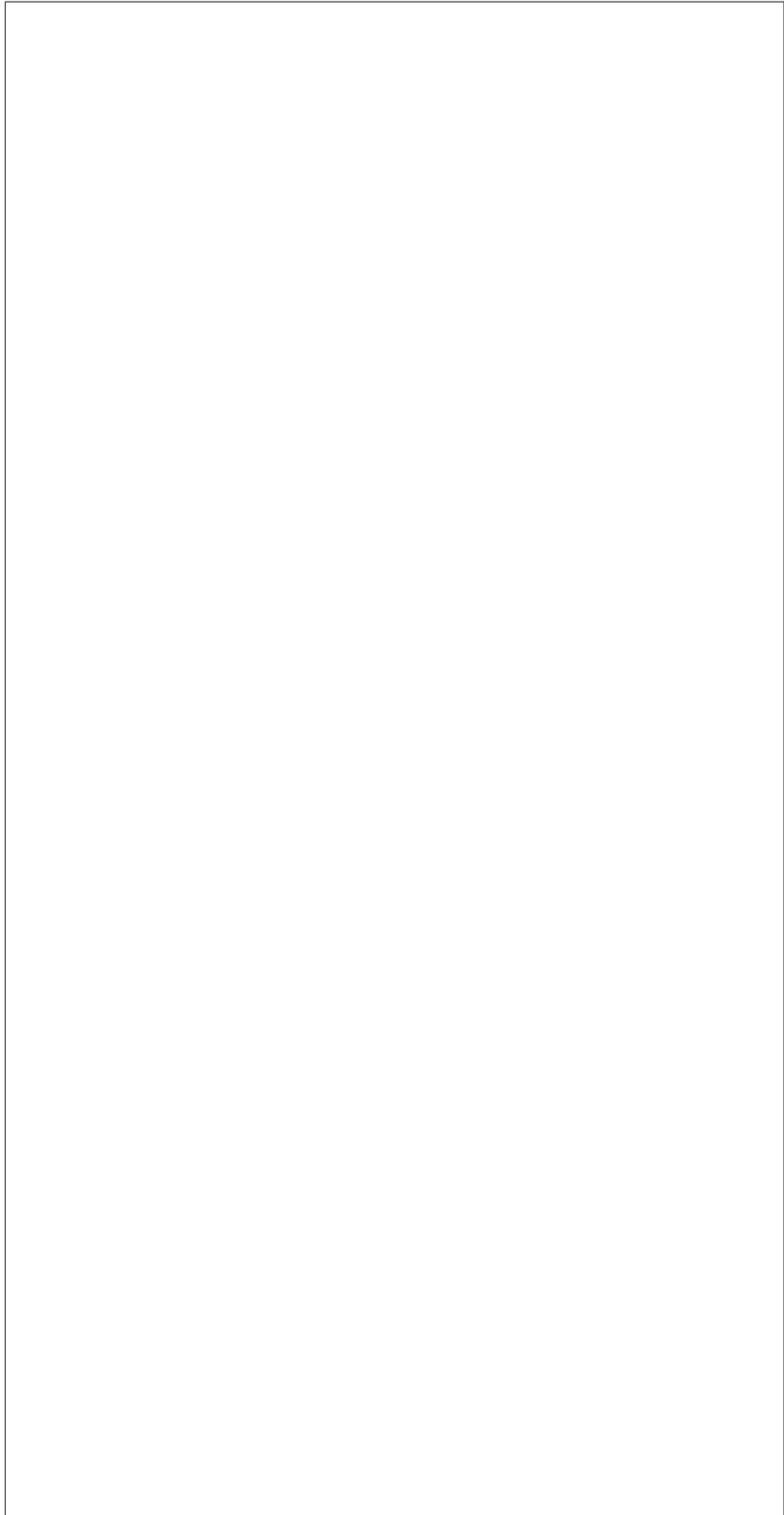
(continues on next page)

(continued from previous page)



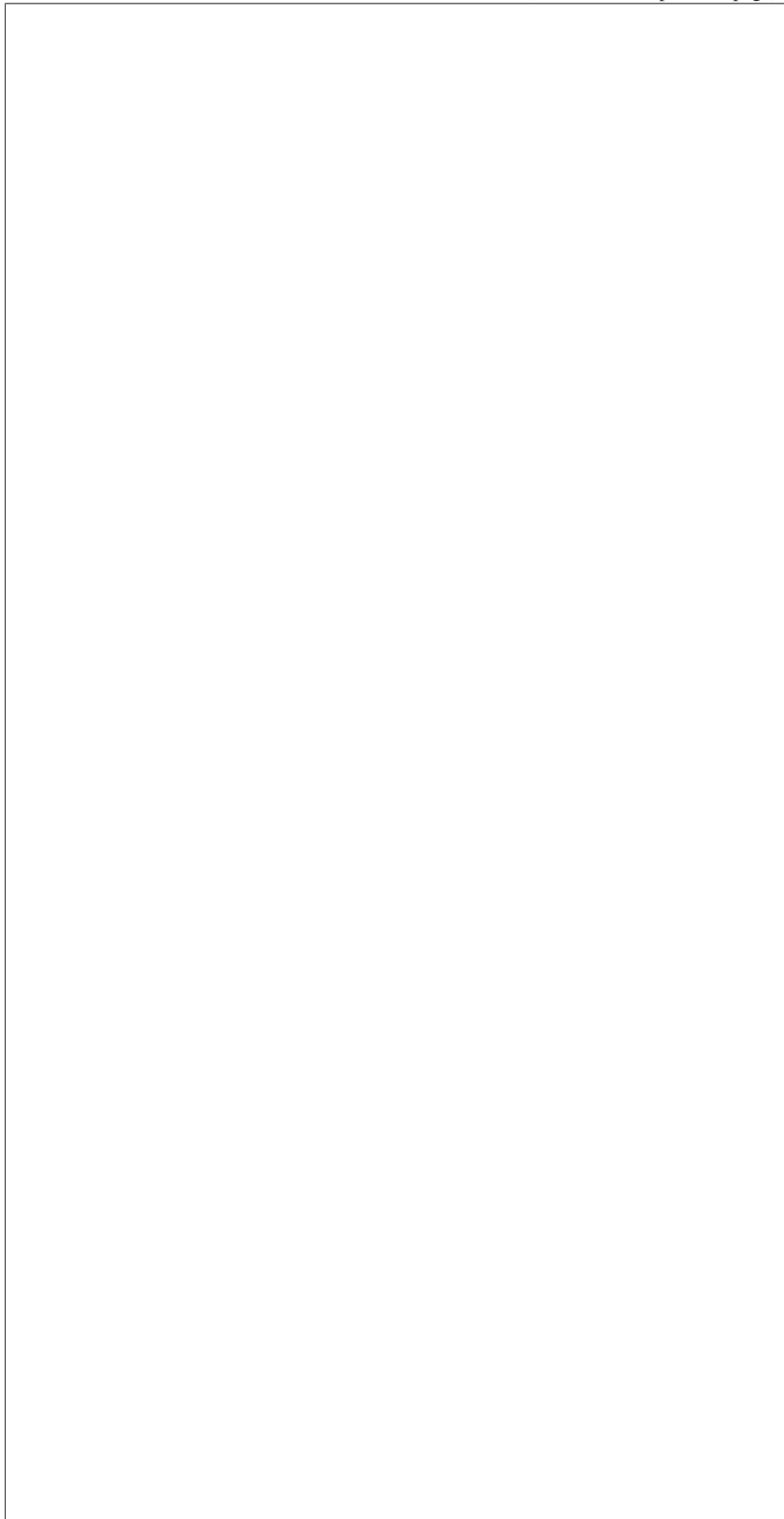
(continues on next page)

(continued from previous page)



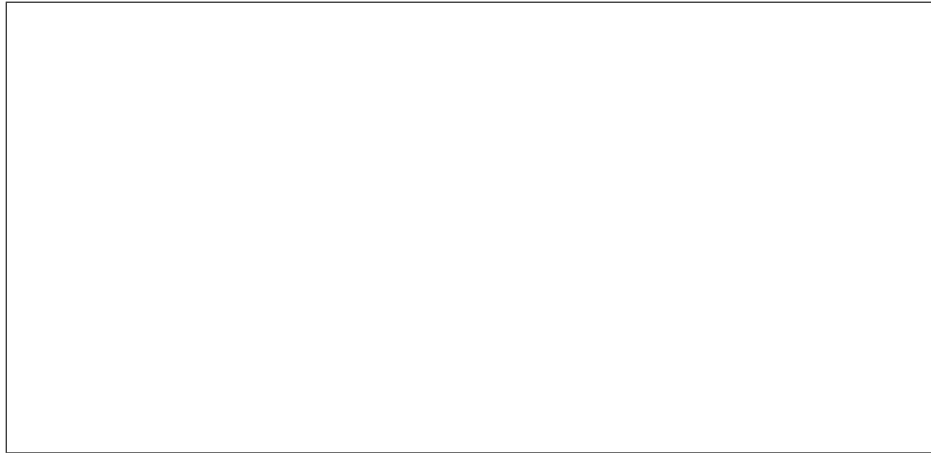
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_r

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Paramet

task
A
Task
ager
in-
stan
con-
tain-

ing
the
node
to
act
on.

Raises

IRM
C-
Op-
er-
a-
tionl
on
an
er-
ror
from
SCC

Returns

Non

restore_

Re-
store
BIO
con-
fig
for
a
node

Paramete

task
a
task
from
Task
ager

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-

e-
cute
step

Returns

Non

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

- **task**
A
task
from
Task
ager

- **dev**
The
boot
de-
vice
one
of
the
sup-

not. Default: False.

port
de-
vice
liste
in
iro.
com
boo

- **per:**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss

ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

set_sec

Set
the
cur-
rent
se-
cure
boot
state
for
the
node

NOTE:

may
not
im-
ple-
men
that.

Parame

- **task**
A task from Task Manager
- **state**
A new state as a boolean

Raises

Missing parameter value if a requirement parameter is missing

Raises

Driver operation is derivative in case of

the hardware

drive
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Drive
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
drive
or

validate

Val-
i-
date
the
drive
spec
man
age-
men
in-
for-
ma-
tion.

This
meth
val-
i-
date
whe
the
drive

information for this driver.

prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Param

task

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
re-
quir
pa-
ram-
e-
ters
are
in-
valic

Raises

Miss
ing-
Pa-

ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.d

Back
BIO
con-
fig
from
a
node

Paramet

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

IRM
C-
Op-
er-
a-
tion
on
fail-
ure.

ironic.drivers.modules.irmc.power module

iRM
Pow
Driv
us-
ing
the
Base
Serv
Pro-
file

class ir

Base
iro
dri
bas
Pow

In-
ter-
face
for
pow
relat
ac-
tions

get_powe

Re-
turn
the
pow
state
of
the
task
node

Parame

task
a
Task
ager
in-
stan
con-
tain-

ing
the
node
to
act
on.

Returns

a
pow
state
One
of
iron
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter

is
miss
ing.

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool
(from
_pow
call)

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list

of
the
sup-
port
pow
state

Parameters

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
cur-
rentl
not
used

Returns

A
list
with
the
sup-
port
pow
state
de-
fine-
in
ironic
comm
sta

reboot (method)

Per-
form
a
hard
re-
boot
of
the

task
node

Parameters

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **time**
time
out
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

indicates default timeout.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if

an
in-
valid
pow
state
was
spec
i-
fied.

Raises

IRM
C-
Op-
er-
a-
tionl
if
faile
to
set
the
pow
state

set_pow

Set
the
pow
state
of
the
task
node

Paramete

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

indicates default timeout.

on.

- **power**
Any
power
state
from
iron
com
sta

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
power
state
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow-
state
was
spec
i-
fied.

the node

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on

Raises
IRM
C-
Op-
er-
a-
tion
if
faile
to
set
the
pow
state

validate
Val-
i-
date
the
driv
spec
Nod
pow
info.
This
meth
val-
i-

information for this driver to manage the power state of the node.

date
when
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Param

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
drive
at-
tribu
is
miss
ing
or

on the node.

in-
valid

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.d

<ironic_

SC2
sc2s
re-
turn
sta-
tus
of
the
cur-
rent
boot

ironic.drivers.modules.irmc.raid module

Irmc
RAI
spe-
cific
meth
ods

class ir

Base
iro

dri
bas
RAI

create_c

Cre-
ate
the
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **cre**

erwise, no root volume is created. Default is True.

ated. Default is True.

If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

Returns
state
if
RAI
con-
fig-
u-
ra-
tion
is

in
prog
asyn
chro

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or
emp

Raises

IRM
C-
Op-
er-
a-
tion
on
an
er-
ror
from
sc-
ci-
clien

delete_c

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

task
a
Task
ager

in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

state
if
dele-
tion
is
in
prog-
asyn-
chro-
or
Non-
if
it
is
com-
plete

get_prop

Re-
turn
the
prop-
er-
ties
of
the
in-
ter-
face

Module contents

`ironic.drivers.modules.network` package

Submodules

`ironic.drivers.modules.network.common` module

```
class ir
    Base
    iro
    dri
    mod
    net
    com
    VIF

VIF
port
ID
mixi
class
for
neu-
tron
net-
worl
in-
ter-
face

Mix
class
that
pro-
vide
VIF
relat
net-
worl
in-
ter-
face
meth
ods
for
neu-
tron
net-
worl
in-
```

terfaces. On VIF attach/detach, the associated neutron port will be updated.

VIFs.

get_node

Get
net-
worl
con-
fig-
u-
ra-
tion
data
for
node
port

Pull
net-
worl
data
from
iron
node
ob-
ject
if
pres
oth-
er-
wise
col-
lect
it
for
Neu
tron

Parame

task
A
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-

is invalid.

ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

Returns

a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
ad-
hear

network metadata layout (*network_data.json*).

ing
Nov

port_cha

Han
dle
any
ac-
tions
re-
quir
whe
a
port
chan

Paramete

- **task**
a
Task
ager
in-
stan

- **port**
a
chan
Port
ob-
ject
from
the
API
be-
fore
it
is
save
to
data

Raises

Fail
ToU
dat-
eD-
HCF
tOn-

Port
Con
flict

portgroup

Han
dle
any
ac-
tions
re-
quir
whe
a
port
grou
chan

Parame

- **task**
a
Task
ager
in-
stan

- **port**
a
chan
Port
grou
ob-
ject
from
the
API
be-
fore
it
is
save
to
data

Raises
Fail
ToU
dat-
eD-

HCF
tOn-
Port
Con
flict

vif_atta

At-
tach
a
vir-
tual
net-
worl
in-
ter-
face
to
a
node

At-
tach
a
vir-
tual
in-
ter-
face
to
a
node

Whe
se-
lect-
ing
a
port
or
port
grou

to attach the virtual interface to, the following ordered criteria are applied:

- Re-
quir
port
or
port
grou
to
have

or one of the VIFs allowed physical networks.

a
phys
i-
cal
net-
worl
that
is
ei-
ther
Non

- Pre-fer port or port group with a phys i-cal net- worl field whic is not Non

- Pre-fer port group to port

- Pre-fer port with PXE en-able

Parame

whose value is a unique identifier for that VIF.

- **task**
A Task manager instance.
- **vif**
a dictionary of information about a VIF. It must have an id key,

Raises
NetworkError, ValueError, AttributeError, NotImplementedError, Port

Raises
PortGroup, PhysicalNetwork, Inconsistency

all assigned the same physical network.

tent
if
one
of
the
node
port
grou
has
port
whic
are
not

vif_det

De-
tach
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

- **task**
A
Task
ager
in-
stan
- **vif**
A
VIF
ID
to
de-
tach

Raises
VifN

tAt-
tach
if
VIF
not
at-
tach

Raises

Net-
worl
Er-
ror
if
un-
bind
Neu
tron
port
faile

class ir

Base
obj
VIF
port
ID
mixi
class
for
non-
neut
net-
worl
in-
ter-
face
Mix
class
that
pro-
vide
VIF
relat
net-
worl
in-
ter-
face
meth
ods

terfaces. There are no effects due to VIF attach/detach that are external to ironic.

vif_attach, vif_detach, port_changed, or portgroup_changed.

for
non-
neut
net-
worl
in-

NOT
This
does
not
yet
sup-
port
the
full
set
of
VIF
meth
ods,
as
it
does
not
pro-
vide

get_curr

Re-
turn
the
cur-
rent
used
VIF
as-
so-
ci-
ated
with
port
or
port
grou

We
are
boot
ing

the
node
only
in
one
net-
work
at
a
time
and
pres
ence
of
clea
ing_

means were doing cleaning, of provisioning_vif_port_id - provisioning, of rescuing_vif_port_id - res-
cuing. Otherwise its a tenant network

Parame

- **task**
A
Task
ager
in-
stan
- **p_ob**
Iron
port
or
port
grou
ob-
ject.

Returns

VIF
ID
as-
so-
ci-
ated
with
p_ob
or
Non

vif_list

List
at-
tach
VIF
IDs
for
a
node

Parameters
task

A
Task
ager
in-
stan

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

ID of the VIF.

ironic.d

Find
free
port
like
ob-
ject
(por
grou
or
port

VIF
will
be
at-
tach
to.
En-
sure
that
the
VIF
is
not
al-
read
at-
tach
to
this
node
When
se-
lect-
ing
a
port

or portgroup to attach the virtual interface to, the following ordered criteria are applied:

- Re-
quir
port
or
port
grou
to
have
a
phys
i-
cal
net-
worl
that
is
ei-
ther
Non
- Pre-

or one of the VIFs allowed physical networks.

fer
port
or
port
group
with
a
phys
i-
cal
net-
work
field
whic
is
not
Non

- Pre-fer port group to port
- Pre-fer port with PXE enable

Paramet

- **task**
a
Task
ager
in-
stan
- **vif**
Nam
or
UUI
of
a

erned by the segments of the VIFs network. An empty set indicates that the ports physical networks should be ignored.

VIF

- **phy:**
Set of physical networks on which the VIF may be attached. This is gov-

- **vif:**
dict that may contain extra information, such as port

Raises
Vi-
fAl-
read
At-
tach
if
VIF
is
al-
read

at-
tach
to
the
node

Raises

NoF
hys-
i-
cal-
Port
if
there
is
no
port
like
ob-
ject
VIF
can
be
at-
tach
to.

Raises

Port
grou
Phys
net-
Inco
sis-
tent
if
one
of
the
node
port
grou
has
port
whic
are
not

all assigned the same physical network.

Returns

port
like
ob-

ject
VIF
will
be
at-
tach
to.

ironic.d

Plug
port
like
ob-
ject
to
ten-
ant
net-
worl

Paramet

- **tasl**
A
Task
ager
in-
stan
- **por**
port
like
ob-
ject
to
plug
- **cli**
Neu
tron
clien
in-
stan

Raises

Net-
worl
Er-

ror
if
faile
to
up-
date
Neu
tron
port

Raises

VifN
tAt-
tach
if
ten-
ant
VIF
is
not
as-
so-
ci-
ated
with
port

ironic.drivers.modules.network.flat module

Flat
net-
worl
in-
ter-
face
Use-
ful
for
shar
flat
net-
worl

class ir

Base
ironic
drivers
modules
network
common

Neu
iro.
com
neu
Neu
iro.
dri
bas
Net
Flat
net-
worl
in-
ter-
face

add_clea

Add
the
clea
ing
net-
worl
to
a
node

Parame tas

A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-

ror,
In-
valid
Pa-
ram-
e-
ter-
Valu

add_insp

Add
the
in-
spec
tion
net-
worl
to
the
node

Parame

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-

is invalid.

ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

add_prov

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Paramete

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror
whe
faile
to
set
bind
ing:

add_reso

Add
the

ure_tenant_network() unbound it.

res-
cu-
ing
net-
worl
to
a
node

Flat
net-
worl
does
not
use
the
res-
cu-
ing
net-
worl
Bind
the
port
agai
sinc
un-
con-
fig-

Parame

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-

world
Er-
ror,
In-
valid
Pa-
ram-
e-
ter-
Valu

configur

Con
fig-
ure
ten-
ant
net-
world
for
a
node

Parame

task
A
Task
ager
in-
stan

remove_c

Re-
mov
the
clea
ing
net-
world
from
a
node

Parame

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

remove_1

Re-
mov
the
in-
spec
tion
net-
worl
from
a
node

Parame

task
A
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

is invalid.

Raises

Miss
ing-
Pa-

ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

remove_p

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Paramet

task
A
Task
ager
in-
stan

remove_r

Re-
mov
the
res-
cu-
ing
net-
worl
from
a
node

Flat
net-
worl

it.

does
not
use
the
res-
cu-
ing
net-
worl
Un-
bind
the
port
agai
sinc
add_
boun

Parame
task

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

unconfi

Un-
con-
fig-
ure
ten-
ant
net-
worl
for
a
node
Un-
bind
the
port
here
to

the tenant and cleaning networks at the same time.

avoi
the
pos-
si-
bil-
ity
of
the
iron
port
be-
ing
boun
to

Parame

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

validat

Val-
i-
date
the
net-
worl
in-
ter-
face

Parame

task
a
Task
ager
in-
stan

Raises

In-
valic
Pa-

is invalid.

`ironic.drivers.modules.network.neutron` module

ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

class ir

Base
iro
dri
mod
net
comm
Neu
iro
comm
neu
Neu

iron
dri
bas
Net

Neu
tron
v2
net-
worl
in-
ter-
face

add_cle

Cre-
ate
neu-
tron
port
for
each
port
on
task
to
boot
the
rame

Parame

task
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

Returns

a
dic-
tio-
nary
in
the
form

{por
neu-
tron

add_insp

Add
the
in-
spec
tion
net-
worl
to
the
node

Parame

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-

is invalid.

world
in-
ter-
face
con-
fig-
u-
ra-
tion

add_prov

Add
the
pro-
vi-
sion
ing
net-
world
to
a
node

Parame

task
A
Task
ager
in-
stan

Raises

Net-
world
Er-
ror

add_res

Cre-
ate
neu-
tron
port
for
each
port
to
boot
the
res-
cue

ram

Parameter

task

a
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

configuration

Con
fig-
ure
ten-
ant
net-
worl
for
a
node

Parameter

task

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

need_power

Che
if
the
node

has
any
Sma
NIC
port

Parame

tas
A
Task
ager
in-
stan

Returns

A
bool
to
in-
di-
cate
Sma
NIC
port
pres
ence

remove_c

Dele
the
neu-
tron
port
cre-
ated
for
boot
ing
the
ram

Parame

tas
a
Task
ager
in-
stan

Raises

Net-
worl
Er-

ror
remove_
Re-
mov
the
in-
spec
tion
net-
worl
from
a
node

Parame
tas
A
Task
ager
in-
stan

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

is invalid.

Raises
Miss
ing-
Pa-
ram-
e-
ter-

Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

remove_p

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parame

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

remove_p

Dele
neu-
tron
port
cre-
ated
for
boot
ing
the
res-
cue

ram

Parameter

task

a

Task

ager

in-

stan

Raises

Net-

work

Er-

ror

unconfig

Un-

con-

fig-

ure

ten-

ant

net-

work

for-

a

node

Nov

take

care

of

port

re-

mov

from

ten-

ant

net-

work

we

un-

bind

it

here

to

avoi

the

possibility of the ironic port being bound to the tenant and cleaning networks at the same time.

Parameter

task

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

validate

Val-
i-
date
the
net-
worl
in-
ter-
face

Parameter

task
a
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

is invalid.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

validate

Val-
i-
date
the
net-
worl
in-
ter-
face
for
res-
cue
op-
er-
a-
tion.

Paramet

task
a
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

is invalid.

ironic.drivers.modules.network.noop module

if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

class ir

Base
iro
dri
bas
Net
Noo
net-
worl
in-
ter-
face

add_clea

Add
the
clear
ing
net-
worl
to
a
node

Parame

task

A
Task
ager
in-
stan

add_prov

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame

task

A
Task
ager
in-
stan

configur

Con
fig-
ure
ten-
ant
net-
worl
for
a
node

Parame

task
A
Task
ager
in-
stand

get_curr

Re-
turn
the
cur-
rent
used
VIF
as-
so-
ci-
ated
with
port
or
port
grou

We
are
boot
ing
the
node
only
in
one
net-
work
at
a
time
and
pres
ence
of
clea
ing_

means were doing cleaning, of provisioning_vif_port_id - provisioning of rescuing_vif_port_id - rescuing. Otherwise its a tenant network

Parame

-

task

A Task object is an instance of the `Task` class.

-

p_obj

IronPython port or port group object.

Returns

VIF ID associated with p_obj or None

port_char

Handle any actions required when a port character

Parameters

-

task
a Task object

- **port**
a
char
Port
ob-
ject.

Raises
Con
flict.
Fail
ToU
dat-
eD-
HCF
tOn-
Port

portgroup

Han
dle
any
ac-
tions
re-
quir
whe
a
port
grou
char

Parame

- **tas**
a
Task
ager
in-
stan

- **por**
a
char
Port
grou
ob-
ject.

Raises

Con
flict.
Fail
ToU
dat-
eD-
HCF
tOn-
Port

remove_c

Re-
mov
the
clea
ing
net-
worl
from
a
node

Paramet

task
A
Task
ager
in-
stan

remove_p

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Paramet

task
A
Task
ager
in-
stan

unconfi

Un-
con-
fig-
ure
ten-
ant
net-
worl
for
a
node

**Parame
task**

A
Task
ager
in-
stan

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

**Parame
task**

A
Task
ager
in-
stan
with
the
node
be-
ing

check

vif_atta

At-
tach
a
vir-
tual
net-
work
worl
in-
ter-
face
to
a
node

Parame

- **task**
A
Task
ager
in-
stan

- **vif**
a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.
It
mus
have
an
id
key,

whose value is a unique identifier for that VIF.

Raises
Net-
worl

Er-
ror,
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-
Port

vif_deta

De-
tach
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

- **task**
A
Task
ager
in-
stan
- **vif**
A
VIF
ID
to
de-
tach

Raises

Net-
worl
Er-
ror,

VifN
tAt-
tach

vif_list

List
at-
tach
VIF
IDs
for
a
node

Parameters

task
A
Task
ager
in-
stan

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

ID of the VIF.

Module contents

`ironic.drivers.modules.redfish` package

Submodules

`ironic.drivers.modules.redfish.bios` module

`class ir`

Base
ironic
drivers
modules
redfish
bios

`apply_co`

Ap-
ply
the
BIO
set-
tings
to
the
node

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **set**
a
list
of

BIO
set-
tings
to
be
up-
date

Raises

Red-
fish-
Con-
nec-
tion-
Erro-
r whe-
n it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush-
li-
brary

cache_b:

Stor-
age
or
up-
date
the
cur-
rent
BIO
set-
tings
for

the
node

Get
the
cur-
rent
BIO
set-
ting
and
store
them
in
the
bios
data
ta-
ble.

Parame

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Red.
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brar

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
sys-
tem
does
not
sup-
port
BIO
set-
tings

factory.

Re-
set
the
BIO
set-
tings
of
the
node
to
the
fac-
tory
de-
fault

Parameter

task

a Task object representing a task in a task manager. The task is contained in a task manager. The task is used for scheduling and execution. The task is used to act on.

Raises

RedfishConnectionError when it fails to connect to Redfish.

Raises

RedfishConnectionError when an error occurs from the SuspectLibrary.

get_properties

Return the

prop-
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro-
erty
nam-
de-
scrip-
tion:
en-
tries

post_con

Per-
form
post
con-
fig-
u-
ra-
tion
ac-
tion
to
store
the
BIO
set-
tings
Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-

override this method to perform a custom action to write the BIOS settings to the Redfish service. The default implementation performs a reboot.

men
ta-
tions
to
ex-
tend
this
class
and

Parameters

- **task_manager**
a TaskManager instance containing the node to act on.
- **settings**
a list of BIOS settings to be updated

post_reboot

Per-
form
post
re-
set
ac-
tion

override this method to perform a custom action to apply the BIOS factory reset to the Redfish service. The default implementation performs a reboot.

to
ap-
ply
the
BIO
fac-
tory
re-
set.
Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-
men-
ta-
tions
to
ex-
tend
this
class
and

Parame

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

validat

Val-
i-

date
the
driv
in-
for-
ma-
tion
need
by
the
red-
fish
driv

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-

Valu
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.drivers.modules.redfish.boot module

class ir

Base
ironic
drivers
modules
redfish
boot

Vir-
tual
me-
dia
boot
in-
ter-
face
over
Red-
fish.

Vir-
tual
Me-
dia
al-
lows
boot
ing
the
sys-
tem
from
the
vir-
tual
CD/
drive
con-
tain-
ing

the user image that BMC inserts into the drive.

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

The
CD/
im-
ages
must
be
in
ISO
for-
mat
and
(de-
pend
ing
on
BMC
im-
ple-
men-
ta-

The
base
line
boot
work
flow
look
like
this:

1. Pull
ker-
nel,
ram
and
ESP
(FA
(FA
par-
ti-
tion
im-
age
with
EFI
boot
load
im-
ages

is only needed for UEFI boot)

Swift temporary URL

push it to Glance and pass to the BMC as Swift temporary URL

- (ES
- 2. Cre-
ate
boot
ISO
out
of
im-
ages
(#1)
push
it
to
Glance
and
pass
to
the
BMC
as
- 3. Op-
tion-
ally
cre-
ate
flopp
im-
age
with
de-
sired
sys-
tem
con-
fig-
u-
ra-
tion
data
- 4. In-
sert
CD/
and
(op-

cue_kernel/rescue_ramdisk properties from *[instance_info]* or *[driver_info]*.

tion-
ally)
flopp
im-
ages
and
set
prop
boot
mod

For
buil
ing
de-
ploy
or
res-
cue
ISO
red-
fish
boot
in-
ter-
face
uses
*de-
ploy*
or
res-

For
buil
ing
boot
(use
ISO
red-
fish
boot
in-
ter-
face
seek
*ker-
nel_*
and
ram
prop
er-

in the Glance image metadata found in `[instance_info]image_source` node property.

ties

capabilities:
`'ramdisk'`

clean_up:

Clean up the boot of in-stance

This method cleans up the environment that was setup for booting the in-stance

Parameters:
task_manager
A task manager object

Returns:
None

clean_up:

Clean up the boot of iron

ram
This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy
ram

Parame

task
A
task
from
Task
ager

Returns

Non

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty

nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
boot
of
in-
stan
over
vir-
tual
me-
dia.

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

tion from the nodes instance_info.

The
in-
ter-
nal
logic
is
as
fol-
lows

- If *boot* re-ques for this de-ploy is lo-cal, then set the node to boot from disk

- Un-less *boot* re-ques for this de-ploy is ram pass root disk ID to vir-tual me-

dia boot image

- Oth-er-wise build boot im-age, in-

node to boot from CD.

sert
it
into
vir-
tual
me-
dia
de-
vice
and
set

Parame

task
a
task
from
Task
ager

Returns

Non

Raises

In-
stan-
ploy
Fail-
ure,
if
its
try
to
boot
iSCS
vol-
ume
in
BIO
boot
mod

prepare.

Pre-
pare
the
boot
of
de-
ploy

vant information from the nodes driver_info and instance_info.

or
res-
cue
ram
over
vir-
tual
me-
dia.
This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

Parame

- **task**
A
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass

or instance_info.

to
the
ram

Returns
Non

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
driv

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises
Iron
icEx
cep-

the node.

tion.
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

validate

Val-
i-
date
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the

the required information for this interface to function.

task
node
con-
tains

Paramete

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parameter

task

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-

eters

thing (default).

ram-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

ironic.d

Ejec
vir-
tual
CDs
and
DVI

Paramet

- **task**
A
task
from
Task
ager
- **boot**
sush
boot
de-
vice
e.g.
VIR-
TUA
VIR-
TUA
or
VIR-
TUA
or
Non
to
ejec
ev-
ery-

the node.

`ironic.drivers.modules.redfish.inspect` module

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
no
suit-
able
vir-
tual
CD
or
DVI
is
foun
on

Red-
fish
In-
spec
In-
ter-
face

class `ir`

Base
iro
dri
bas
Ins

get_prop

Re-
turn
the
prop
er-
ties
of
the

in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect.

In-
spec
hard
ware
to
get
the
hard
ware
prop
er-
ties.

In-
spec
hard
ware
to
get
the
es-
sen-
tial
prop
er-
ties.
It
fails
if
any
of
the
es-

sential properties are not received from the node.

cessfully.

Parame

tasl

a

Task

ager

in-

stan

Raises

Har

war

spec

tion-

Fail-

ure

if

es-

sen-

tial

prop

er-

ties

coul

not

be

re-

triev

suc-

Returns

The

re-

sult-

ing

state

of

in-

spec

tion.

validate

Val-

i-

date

the

driv

spec

Nod

de-

information for this interface to function.

long-running checks.

ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
prop
er-
ties
of
the
task
node
con-
tain
the
re-
quir

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Paramet
task
A
Task
ager

in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

`ironic.drivers.modules.redfish.management` module

class `ir`

Base
ironic
drivers
base
Management

clear_se

Clear
all
se-
cure
boot
keys

Parameter

task
a
task
from
Task
manager

Raises

Un-
sup-
port
ed-
Drive
ten-
sion
if
se-
cure
boot
is
now
sup-
port

Raises

Red-
fish-
Er-
ror
on
run-
time

drive
er-
ror.

detect_v

De-
tects
and
re-
turn
the
hard
ware
ven-
dor.

Uses
the
Sys-
tems
Man-
u-
fac-
ture
field

Paramete

task
A
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com-
po-
nent
in-
di-
ca-

state is specified.

tor
or

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Red
fish-
Er-
ror
on
driv
spec
prob
lems

Returns

Strin
rep-
re-
sent
ing
the
BM
re-
port
Ven-
dor
or
Man
u-
fac-
ture
oth-

returns None.

er-
wise

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Parameter

task
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-

ram-
e-
ter(s)

Raises

Red-
fish-
Con-
nec-
tion-
Erro-
whe-
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brar

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of

unknown.

iron
com
boo
or
Non
if
it
is
un-
know

persist
Boo
valu
or
Non
True
if
the
boot
de-
vice
per-
sists
Fals
oth-
er-
wise
Non
if
its

get_boot
Get
the
cur-
rent
boot
mod
for
a
node
Pro-
vide
the
cur-
rent
boot
mod
of

the
node

Parameters

task

A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Driv
Op-
er-
a-
tionl
or
its
deri
tive
in
case
of
driv
run-
time
er-
ror.

Returns

The
boot

mod
one
of
iron
comm
boo
or
Non
if
it
is
un-
know

get_ind:

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard
ware
com
po-
nent

Paramete

- **task**
A
task
from
Task
ager
- **comp**
The
hard
ware
com
po-
nent

one
of
iro
comm
comp

- **ind.**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises
Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brary

Returns

Current state of the indicated capacitor, one of [iron](#), [common](#), [inductor](#)

get_mac

Get MAC address information for the node

Parameter

task
A Task object representing an instance containing the node to act on.

Raises

RedfishConnectionError

when
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush-
li-
brary

Returns

a
dic-
tio-
nary
con-
tain-
ing
MA-
ad-
dres-
of
en-
able
in-
ter-
face
in
a
{ma

state} format

get_prop

Re-
turn
the
prop

er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_secu

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

Parame

task
A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if

a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Red-
fish-
Er-
ror
or
its
deri
tive
in
case
of
a
drive
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Drive
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
hard
ware

Returns

Boo

get_sens

Get
sen-
sors
data

Param

task
a
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data
fails

Raises

Fail
ToP
eSer
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises

In-
vali
Pa-
ram
e-
ter-
Valu

if
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

re-
turn
a
dict
of
sen-
sor
data
grou
by
sen-
sor
type

get_supp

Get
a
list
of
the

sup-
port
boot
de-
vice

Parameters
task
a
task
from
Task
ager

Returns
A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
[ironic](#)
[command](#)
[book](#)

get_support
Get
a
list
of
the
sup-
port
boot
mod

Parameters
task
A
task
from
Task
ager

Returns
A
list

can't be determined, empty list is returned.

with
the
sup-
port
boot
mod
de-
fine
in
iro.
com
boo
If
boot
mod
sup-
port

get_sup

Get
a
map
of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Param

- **task**
A
task
from
Task
ager
- **comp**
If
not
Non
re-
turn

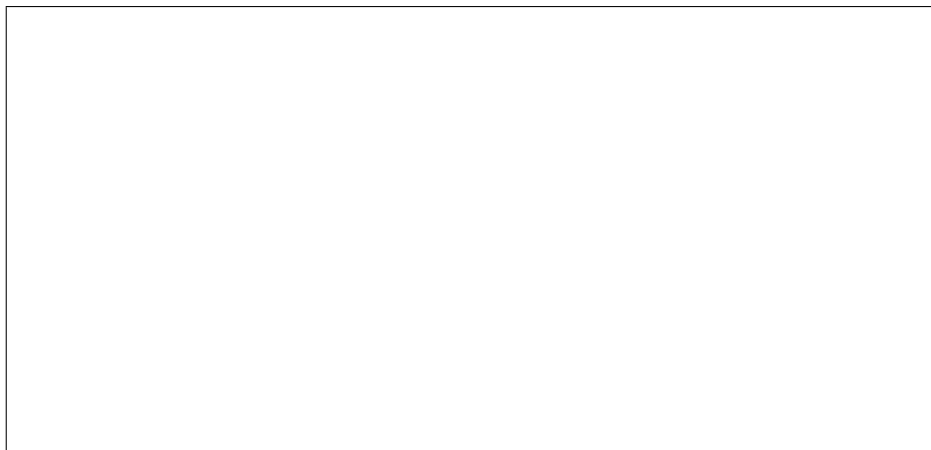
in-
di-
ca-
tor
in-
for
ma-
tion
for
just
this
com
po-

nent, otherwise return indicators for all existing components.

Returns

A
dic-
tio-
nary
of
hard
ware
com-
po-
nent
(*ironic*
components
components
as
keys
with
val-
ues
be-
ing

dictionaries having indicator IDs as keys and indicator properties as values.



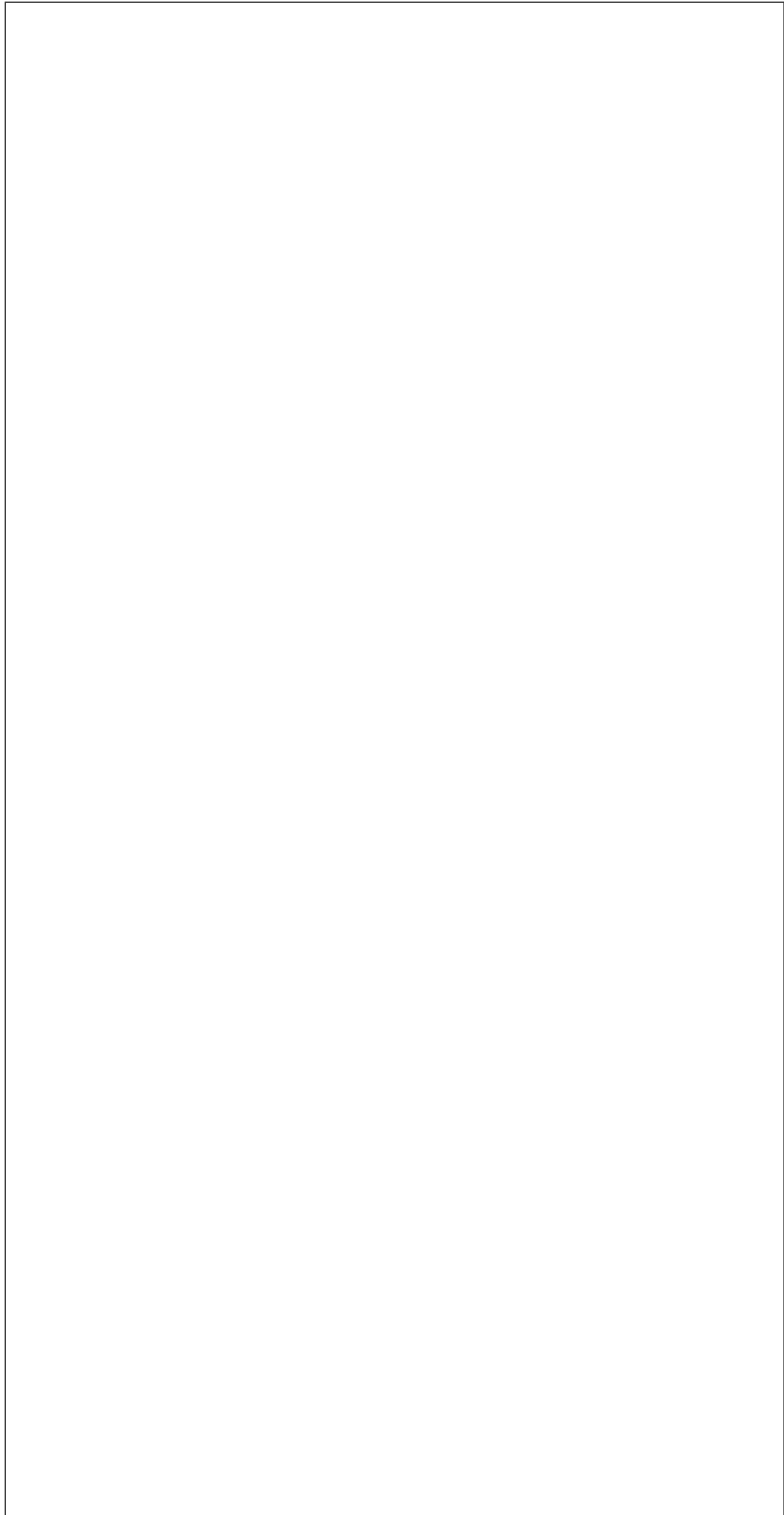
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



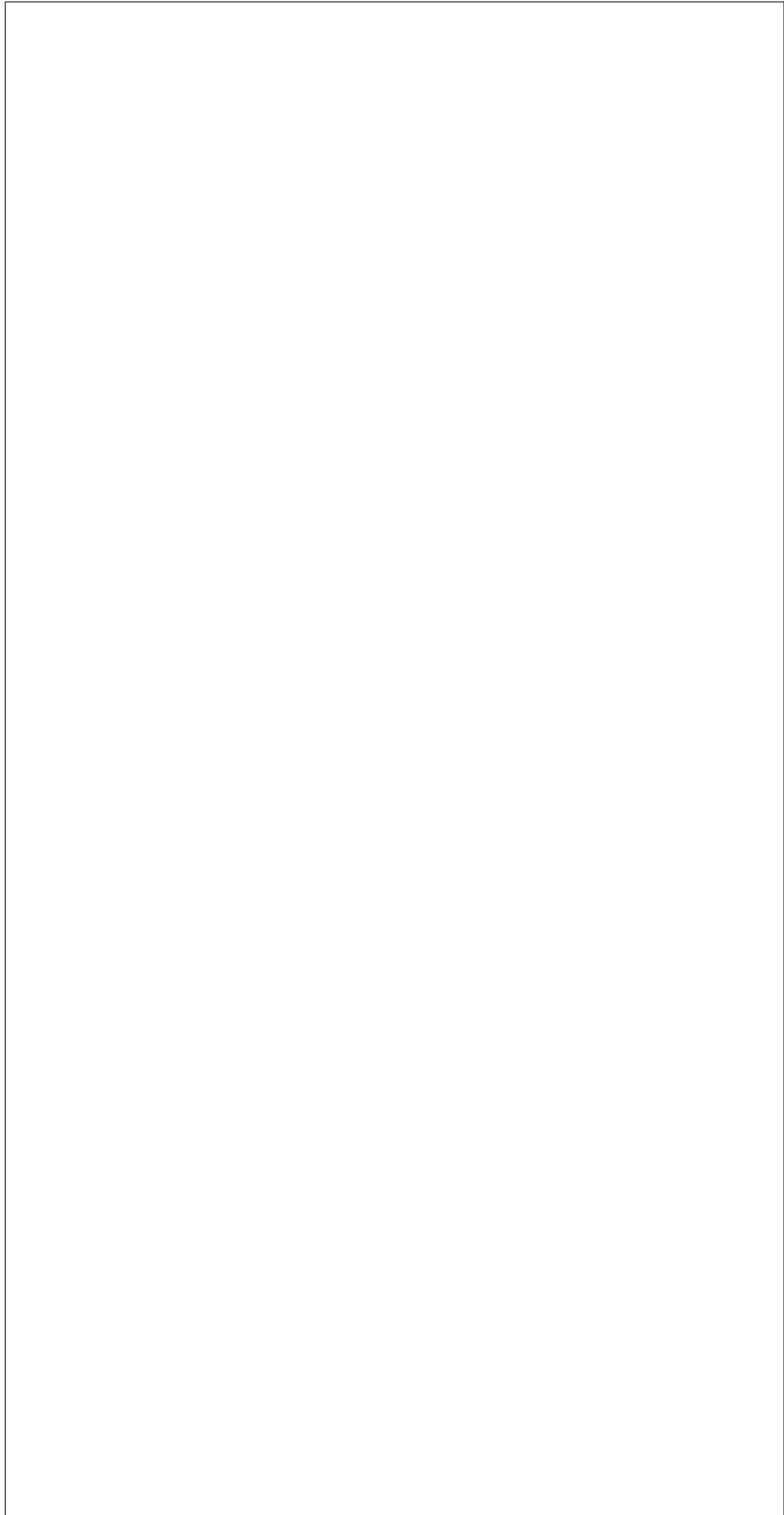
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



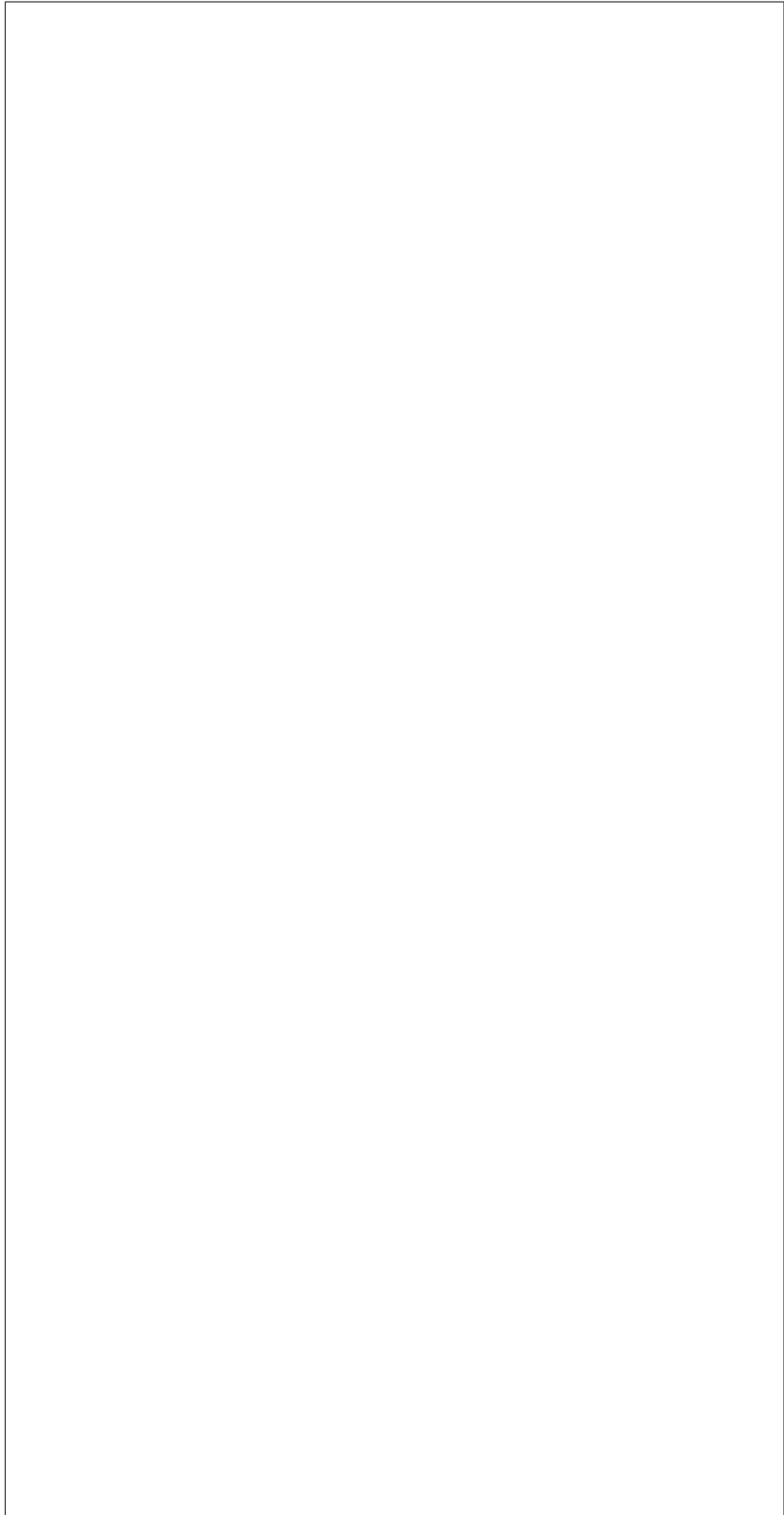
(continues on next page)

(continued from previous page)



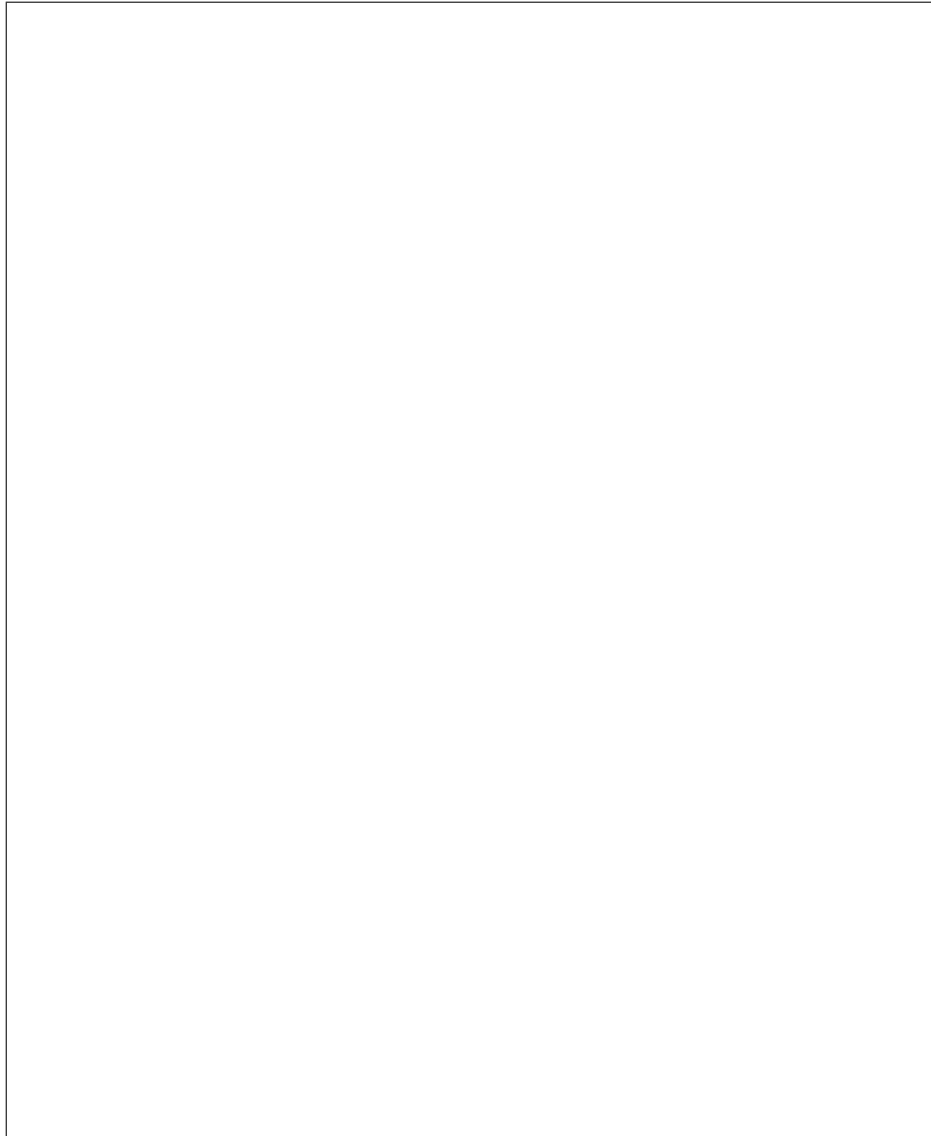
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_r

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-

ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing

pa-
ram-
e-
ter(s)

Raises

Red-
fish-
Con-
nec-
tion-
Erro-
r whe-
n it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sus-
li-
brary

reset_s-

Re-
set
se-
cure
boot
keys
to
man-
u-
fac-
tur-
ing
de-
fault

Parame

tas

a

task

from

Task

ager

Raises

Un-

sup-

port

ed-

Driv

ten-

sion

if

se-

cure

boot

is

now

sup-

port

Raises

Red

fish-

Er-

ror

on

run-

time

drive

er-

ror.

restore_

Re-

store

boot

de-

vice

if

need

Che

the

red-

fish_

in-

ter-

warning is issued if it fails.

sidered private to the Redfish hardware type.

nal
flag
and
sets
the
one-
time
boot
de-
vice
ac-
cord
ingly
A

This
meth
is
sup-
pose
to
be
calle
from
the
Red-
fish
pow
in-
ter-
face
and
shou
be
con-

Paramet

- **task**
a
task
from
Task
ager
- **sys**
a
Red-

fish
Sys-
tem
ob-
ject.

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parameter

- **task**
a
task
from
Task
ager
- **dev**
the
boot
de-
vice
one
of
[ironic](#)
[command](#)
[boot](#)

not. Default: False.

- **per-**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ter(s)

Raises

RedfishConnectionError when it fails to connect to Redfish

Raises

RedfishError on an error from the Sushli-brary

set_boot

Set the boot mode for a node

Set the boot mode to use on next re-

boot
of
the
node

Parameter

- **task**
A
task
from
Task
ager

- **mode**
The
boot
mod
one
of
ironic
command
boot

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Red-
fish-
Con-
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brar

set_ind:

Set
in-
di-
ca-
tor
on
the

hard
ware
com
po-
nent
to
the
de-
sired
state

Paramete

- **task**
A
task
from
Task
ager

- **comp**
The
hard
ware
com
po-
nent
one
of
ironic
com
com

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

- **sta**
De-
sired
state

of
the
in-
di-
ca-
tor,
one
of
iron
com
ind

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
in-
di-
ca-
tor
or

state is specified.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss

ing

Raises

Redfish-
Error-
on
an
error-
from
the
Sushli-
brary

set_secu

Set
the
current
secure
boot
state
for
the
node

Parameter

- **task**
A task from Task Manager
- **state**
A new state as a boolean

Raises

Missing-

Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Red-
fish-
Er-
ror
or
its
deri
tive
in
case
of
a
driv
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not
sup-
port

by
the
hard
ware

update_:

Up-
date
the
firm
on
the
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **firm**
A
list
of
firm
im-
ages
are
to
ap-
ply.

Returns

Non-
if
it
is
com-
plete

Raises

Redfish-Error on an error from the Sushli-brar

validate

Validating the driver information needed by the redfish driver

Parameter

task
a Task agent instance containing the node to act on.

Raises

InvalidParameter

ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

`ironic.drivers.modules.redfish.power` module

class `ironic.drivers.modules.redfish.power`

Base
ironic.drivers.modules.redfish.power
Power

get_power

Get
the
cur-
rent
pow
state
of
the
task
node

Parameters

task

a

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Returns

a

pow

state

One

of

iron

comm

sta

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

on

mal-

form

pa-

ram-

e-

ter(s)

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

on

miss

ing

pa-
ram-
e-
ter(s)

Raises

Red-
fish-
Con-
nec-
tion-
Erro-
r whe-
n it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sus-
li-
brary

get_prop

Re-
turn
the
prop-
er-
ties
of
the
in-
ter-
face

Returns

dic-

tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list
of
the
sup-
port
pow
state

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
Not
used
by
this
driv

at the moment.

Returns

A
list
with
the
sup-
port

pow
state
de-
fine
in
iro
com
sta

reboot()

Per-
form
a
hard
re-
boot
of
the
task
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **time**
Tim
to
wait
for
the
node
to
be-
com
pow

ered
on.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

Red-
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-

bran

set_power

Set the power state of the task node

Parameters

- **task_id**
a Task agent instance containing the node to act on.
- **power**
Any power state from [ironic-compute-standby](#)
- **timeout**
Time to wait for the node to reach the re-

ques
state

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

Red-
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-

bran

validate

Val-
i-
date
the
driv
in-
for-
ma-
tion
need
by
the
red-
fish
driv

Parame

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
vali
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

`ironic.drivers.modules.redfish.raid` module

`class ir`

Base
iro
dri
bas
RAI

`apply_co`

Ap-
plies
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
A
Task
ager
in-
stan
-

ified in `raid_config`. Default value is `True`.

raid
The
RAID
con-
fig-
u-
ra-
tion
to
ap-
ply.

- **create**
Set-
ting
this
to
`False`
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **create**
Set-
ting
this
to
`False`
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valic

Returns
state
if
RAI
con-
fig-
u-

plete.

will create all logical disks.

ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_

Cre-
ate
RAI
con-
fig-
u-
ra-
tion
on
the
node

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
as
read
from
node
This
meth
by
de-
fault

Parame

ified in the nodes target_raid_config. Default value is True.

- **task**
Task
ager
ob-
ject
con-
tain-
ing
the
node

- **create**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **create**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume

cept the root volume) in the nodes target_raid_config. Default value is True.

creating the new configuration. Default is False.

plete.

(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

Raises

Red
fish-
Er-
ror

if
there
is
an
er-
ror
cre-
at-
ing
the
con-
fig-
u-
ra-
tion

delete_c

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
node

Parame

task
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion

it is complete.

is
in
prog
asyn
chro
or
Non
if

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

post_cre

Per-
form
post
cre-
ate_
ac-
tion
to
com
mit
the
con-
fig.

override this method to perform a custom action to commit the RAID create configuration to the Redfish service.

Ex-
ten-
sion
poin
to
al-
low
ven-
dor
im-
ple-
men-
ta-
tions
to
ex-
tend
this
class
and

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **raid**
a
list
of
dic-
tio-
nar-
ies
con-
tain-

eration details.

ing
the
RAI
con-
fig-
u-
ra-
tion
op-

- **ret**
state
to
re-
turn
base
on
op-
er-
a-
tion
be-
ing
in-
voke

post_de

Per-
form
post
dele
ac-
tion
to
com
mit
the
con-
fig.
Ex-
ten-
sion
poin
to
al-
low
ven-
dor
im-

override this method to perform a custom action to commit the RAID delete configuration to the Redfish service.

ple-
men-
ta-
tions
to
ex-
tend
this
class
and

Parameters

- **task**
a Task Manager instance containing the node to act on.

- **raid**
a list of dictionaries containing the RAID configuration operation

eration details.

-

retu
state
to
re-
turn
base
on
op-
er-
a-
tion
be-
ing
in-
voke

pre_crea

Per-
form
re-
quir
ac-
tions
be-
fore
cre-
at-
ing
con-
fig.

Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-
men-
ta-
tions
to
ex-
tend
this
class

override this method to perform custom actions prior to creating the RAID configuration on the Redfish service.

and

Parameters

- **task_id**
a Task Manager instance containing the node to act on.
- **log_id**
list of logical disks to create.

pre_delete

Perform required actions before deleting configuration. Extension point

override this method to perform custom actions prior to deleting the RAID configuration on the Redfish service.

to
al-
low
ven-
dor
im-
ple-
men-
ta-
tions
to
ex-
tend
this
class
and

Paramete

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **vol**
list
of
vol-
ume
to
dele

validate

Val-
i-
date
the

RAI
In-
ter-
face

This
meth
val-
i-
date
the
prop
er-
ties
de-
fine
by
Iron
for
RAI
con-
fig-
u-
ra-
tion.

Driver implementations of this interface can override this method for doing more validations (such as BMCs credentials).

Parame

task
A
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is

in-
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

volume_c

Han
dle
er-
ror
from
faile
Vol-
ume
lec-
tion.
Ex-
ten-
sion
poin
to
al-
low
ven-
dor
im-
ple-
men
ta-
tions
to
ex-
tend
this
class

override this method to perform a custom action if the call to `VolumeCollection.create()` fails.

Parameters

- **task**: a TaskManager instance containing the node to act on.
- **exception**: the exception raised by VolumeCollection.
- **volume**: the sushy VolumeCollection instance.
- **payload**: the payload passed to the

faile
cre-
ate()

Returns

New
cre-
ated
Vol-
ume
re-
sour
or
Task
i-
tor
if
asyn
task

Raises

Red
fish-
Er-
ror
if
there
is
an
er-
ror
cre-
at-
ing
the
vir-
tual
disk

`ironic.d`

Con
vert
size
in
log-
i-
cal_
from
gb
to
byte

ironic.d

Cre-
ate
a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

Paramet

- **task**
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **rai**
id
of
the
RAI
con-
troll
- **phys**
ids
of
the
phys
i-
cal
disk

- **raid**
RAID
level
of
the
vir-
tual
disk
- **size**
size
of
the
vir-
tual
disk
- **disk**
name
of
the
vir-
tual
disk
(option)
- **span**
Number
of
of
span
in
vir-
tual
disk
(option)
- **span**
Number
of
of
disk
per
span
(option)

-

erro
func
tion
to
call
if
vol-
ume
cre-
ate
fails
(op-
tion:

Returns

New
cre-
ated
Vol-
ume
re-
sour
or
Task
i-
tor
if
asyn
task

Raises

Red-
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish.

Raises

Red-
fish-
Er-
ror

if
there
is
an
er-
ror
cre-
at-
ing
the
vir-
tual
disk

`ironic.d`

Get
the
phys-
i-
cal
drive
of
the
node

Paramet

node
an
iron-
node
ob-
ject.

Returns

a
list
of
Drive
ob-
jects
from
sush

Raises

Red-
fish-
Con-
nec-
tion-
Erro
whe
it

fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
if
there
is
an
er-
ror
get-
ting
the
drive
via
Red-
fish

ironic.drivers.modules.redfish.utils module

class ir

Base
obj

Cach
of
HTT
ses-
sion
cre-
den-
tials

AUTH_CLA

ironic.d

Get
in-
for-
ma-

tion
on
MA
ad-
dres
of
en-
able
port
us-
ing
Red.
fish.

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **sys**
a
Red.
fish
Sys-
tem
ob-
ject

Returns

a
dic-
tio-
nary
con-
tain-
ing
MA
ad-

state} format

dres
of
en-
able
in-
ter-
face
in
a
{ma

ironic.d

Get
a
Red-
fish
Sys-
tem
that
rep-
re-
sent
a
node

Paramet

nod
an
Iron
node
ob-
ject

Raises

Red-
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Redfish-
Error-
if the
System
is not
registered
in Redfish

ironic.d

Get a Task
i-
tor
for a
node

Parameter

- **node**
an Iron
node ob-
ject
- **uri**
the URI
of a
Task i-
tor

Raises
Redfish-
Con-
nec-

tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
whe
the
Task
i-
tor
is
not
avai
able
in
Red-
fish

ironic.d

Get
a
node
up-
date
ser-
vice

Paramet

node
an
Iron
node
ob-
ject

Raises

Red-
fish-
Con
nec-

tion-
Erro
whe
it
fails
to
con-
nect
to
Red-
fish

Raises

Red-
fish-
Er-
ror
whe
the
Up-
date
Ser-
vice
is
not
reg-
is-
tere
in
Red-
fish

ironic.d

Pars
the
in-
for-
ma-
tion
re-
quir
for
Iron
to
con-
nect
to
Red-
fish.

**Paramet
nod**

an
Iron
node
ob-
ject

Returns

dic-
tio-
nary
of
pa-
ram-
e-
ters

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.d

Wai
un-
til

Redfish system is read

Parameter node
an Iron node object

Raises
RedfishConnectionError on timeout.

`ironic.drivers.modules.redfish.vendor` module

Vendor interface for Redfish driver and its supporting methods.

class `ironic.drivers.modules.redfish.vendor`

Base class for `ironic.drivers.modules.redfish.vendor`

bas
Ven
Ven
spec
in-
ter-
face
for
Red
fish
drive

eject_virt

Ejec
a
vir-
tual
me-
dia
de-
vice

Parame

- **task**
A
Task
ager
ob-
ject.
- **kwargs**
The
ar-
gu-
men-
sent
with
ven-
dor
pass
The
op-
tion:
kwargs
are::
boot
the

vice to eject

boot
de-

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validate

Val-
i-
date
vend
spec
ac-
tions

Che
if
a
valid
ven-
dor
pass
meth
was
pass
and
val-

the vendor passthru method.

i-
date
the
pa-
ram-
e-
ters
for

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **method**
meth
to
be
val-
i-
date
- **kwargs**
kwa
con-
tain-
ing
the
ven-
dor
pass
meth
ods
pa-
ram-
e-

ters.
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

Module contents

`ironic.drivers.modules.storage` package

Submodules

`ironic.drivers.modules.storage.cinder` module

class `ironic.drivers.modules.storage.cinder.CinderStorageDriver`
Base
ironic.drivers.modules.storage.cinder.CinderStorageDriver
A
stor-
age_
driv
sup-
port
ing
Cin-
der.

attach_v

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

task

The
task
ob-
ject.

Raises

Stor-
ageE
ror
If
an
un-
der-
ly-
ing
ex-
cep-
tion
or
fail-
ure
is
de-
tecte

detach_v

In-
form
the
stor-
age

sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

This
ac-
tion
is
re-
tried
in
case
of
fail-
ure.

Paramete

- **task**
The
task
ob-
ject.
- **conn**
The
dic-
tio-
nary
rep-
re-
sent
ing
a
node
con-
nec-
tiv-
ity
as
de-

`_generate_connector()`. Generated if not passed.

a failed attachment

fine
by

- **abon**
Boo
rep-
re-
sent
ing
if
this
de-
tach
men
was
re-
ques
to
han-
dle
abon
ing

Raises

Stor
ageE
ror
If
an
un-
der-
ly-
ing
ex-
cep-
tion
or
fail-
ure
is
de-
tecte

get_prop

Re-
turn
the
prop
er-

ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

should_v

De-
ter-
mine
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

Parame

task
The
task
ob-
ject.

Returns

True
if
the
de-
ploy
men
writ

out
pro-
cess
shou
be
ex-
e-
cute

validate

Val-
i-
date
stor-
age_
con-
fig-
u-
ra-
tion
for
Cin-
der
us-
age.

In
or-
der
to
pro-
vide
fail
fast
func-
tion-
al-
ity
prio-
to
node
be-
ing
re-
ques-
to

enter the active state, this method performs basic checks of the volume connectors, volume targets, and operator defined capabilities. These checks are to help ensure that we should have a compatible configuration prior to activating the node.

Parame
task

The
task
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
If
a
mis-
con-
fig-
u-
ra-
tion
or
mis-
matc
ex-

ists that would prevent storage the cinder storage driver from initializing attachments.

ironic.drivers.modules.storage.external module

class ir

Base
iro
dri
bas
Sto

Ex-
ter-
nally
driv
Stor
age
In-
ter-
face

attach_v

In-
form

the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

task

A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

detach_v

In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parameter

task

A

Task

ager

in-

stan

Raises

Un-

sup-

port

ed-

Driv

ten-

sion

get_prop

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

should_v

De-

ter-

min

if

de-

ploy

shou

may already exist and we may be booting to that volume.

per-
form
the
im-
age
writ
out.
This
en-
able
the
user
to
de-
fine
a
vol-
ume
and
Iron
un-
der-
stan
that
the
im-
age

Parame
task
The
task
ob-
ject.

Returns
True
if
the
de-
ploy
men
writ
out
pro-
cess
shou
be
ex-
e-
cute

the required information for this interface to function.

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
nod
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it

long-running checks.

shou
not
con-
duct

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.drivers.modules.storage.noop module

class ir

Base
iro
dri
bas
Sto
No-
op
Stor
age
In-
ter-
face

attach_v

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

task
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv

ten-
sion

detach_v

In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parame

task
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

should_v

De-
ter-
min
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

Paramet

task
A
Task
ager
in-
stan

Returns

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect

ten by Ironic.

the
im-
age
to
be
writ

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

the required information for this interface to function.

long-running checks.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

Module contents

`ironic.drivers.modules.xclarity` package

Submodules

`ironic.drivers.modules.xclarity.common` module

`ironic.d`

`ironic.d`

Val-
i-
date
node
con-
fig-
u-
ra-
tion
and
re-
turn
xcla
ity
hard
ware
id.
Val-
i-
date

ity Hardware ID for a specific node. :param node: node object to get information from :returns: the XClarity Hardware ID for a specific node :raises: MissingParameterValue if unable to validate XClarity Hardware ID

when
node
con-
figu-
tion
is
con-
sis-
tent
with
XCl
ity
and
re-
turn
the
XCl

ironic.d

Gen
er-
ates
an
in-
stan-
of
the
XCl
ity
clien

Gen
er-
ates
an
in-
stan-
of
the
XCl
ity
clien
us-
ing
the
im-
port
xcla

ity_
li-
brar

Parameter

node
an
iron
node
ob-
ject.

Returns

an
in-
stan
of
the
XCL
ity
clien

Raises

XCL
i-
ty-
Er-
ror
if
cant
get
to
the
XCL
ity
clien

ironic.d

Pars
a
node
driv
val-
ues.

Pars
the
driv
of
the
node
read
de-

combination of both.

fault
val-
ues
and
re-
turn
a
dict
con-
tain-
ing
the

Parameter

node
an
iron
node
ob-
ject
to
get
in-
for-
mati
from

Returns

a
dict
con-
tain-
ing
in-
for-
ma-
tion
pars
from
driv

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
re-

the node or inputs is invalid.

quir
in-
for-
ma-
tion
is
miss
ing
on

ironic.d

Tran
lates
iron
ics
pow
ac-
tion
strin
to
XCL
i-
tys
for-
mat.

Paramet

pow
pow
ac-
tion
strin
to
be
trans
latec

Returns

the
pow
ac-
tion
trans
latec

ironic.d

Tran
lates
XCL
i-
tys

pow
state
strin
to
be
con-
sis-
tent
with
Iron

Parameter

power
power
state
strin
to
be
trans
latec

Returns

the
trans
latec
pow
state

`ironic.drivers.modules.xclarity.management` module

class `ironic.drivers.modules.xclarity.management`

Base
ironic.drivers.modules.xclarity.management
Management

get_boot

Get
the
cur-
rent
boot
de-
vice
for
the
task
node

Parameters
task
a
task
from
Task
ager

Returns
a
dic-
tio-
nary
con-
tain-
ing:
:boot
the
boot
de-
vice
one
of
[PX
DIS
CDE
BIO
:per-

sistent: Whether the boot device will persist or not It returns None if boot device is unknown.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
boot
de-
vice
is
un-
know

Raises
XCl
i-
ty-
Er-
ror

if
the
com
mu-
ni-
ca-
tion
with
XCL
ity
fails

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sens

Get
sen-
sors
data

Paramet

task
a
Task
ager
in-
stan

Raises

NotImplementedError

get_supported_devices

Gets a list of the supported boot devices

Parameter

task
a task from TaskManager

Returns

A list with the supported boot devices defined in [ironic.common.boot](#)

set_boot_device

Sets the boot device for

a
node

Parameter

- **task**
a
task
from
Task
ager
- **dev:**
the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
iron
com
boo
- **per:**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

not. Default: False.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

XCL
i-
ty-
Er-
ror
if
the
com
mu-
ni-
ca-
tion
with
XCL
ity
fails

validate

Val-
i-
date
the
drive
spec
info
sup-
plie
This

quired information for this driver to manage the node.

`ironic.drivers.modules.xclarity.power` module

meth
val-
i-
date
if
the
driv
prop
erty
of
the
sup-
plied
task
node
con-
tains
the
re-

Paramete
task
a
task
from
Task
ager

class ir

Base
ironic
drivers
base
Power

get_powe

Gets
the
cur-
rent
pow
state

Paramete
task
a

Task
ager
in-
stan

Returns

one
of
iro
com
sta
POV
POV
or
ER-
ROF

Raises

XCL
i-
ty-
Er-
ror
if
fails
to
re-
triev
pow
state
of
XCL
ity
re-
sour

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-

nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

reboot (/

Soft
re-
boot
the
node

Paramete

- **task**
a
Task
ager
in-
stan

- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

set_powe

Turn
the
cur-
rent
pow
state

on
or
off.

Parameters

-

task
a
Task
ager
in-
stan

-

power
The
de-
sired
power
state
POV
POV
or
RE-
BOC
from
ironic
command
state

-

time
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

In-
valid
Pa-
ram-
e-

ter-
Valu
if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises

XCL
i-
ty-
Er-
ror
if
XCL
ity
fails
set-
ting
the
pow
state

validate

Val-
i-
date
the
drive
spec
info
sup-
plied

This
meth
val-
i-
date
if
the
drive
prop
erty
of
the

quired information for this driver to manage the power state of the node.

Module contents

Submodules

`ironic.drivers.modules.agent module`

sup-
plied
task
node
con-
tains
the
re-

Paramete
task
a
task
from
Task
ager

class ir

Base
iro
dri
mod
age
Cus

In-
ter-
face
for
depl
relat
ac-
tion:

prepare_

Pre-
pare
in-
stan
for
boot
ing.

The
base
ver-
sion
only
calls
pre-
pare
on
the
boot
in-
ter-
face

validate

Val-
i-
date
the
drive
spec
Node
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
prop
er-
ties
of
the
sup-
plied
node
con-
tain
the
re-
quir

information for this driver to deploy images to the node.

**Paramet
task**

ing.

a
Task
ager
in-
stand

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

write_in

class ir

Base
iron
driver
base
RAI
Im-
ple-
men-
ta-
tion
of
RAI
In-
ter-
face
which
uses
ager
rame

apply_c

Ap-
plies
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-

creating the new configuration.

fig-
u-
ra-
tion
to
ap-
ply.

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valic

Returns
state
if

plete.

RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_

Cre-
ate
a
RAI
con-
fig-
u-
ra-
tion
on
a
bare
meta
us-
ing
agen
ram

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the

give
node

Parameter

- **task**
a
Task
ager
in-
stan
- **create**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

erwise, no root volume is created. Default is True.

- **create**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-

ated. Default is True.

ter skipping root volume and/or non-root volumes.

ume
are
cre-

Returns

state
if
op-
er-
a-
tion
was
suc-
cess
fully
in-
voke

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or
was
foun
to
be
emp
af-

delete_c

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give

node

Parameter

task

a

Task

ager

in-

stan

Returns

state

if

op-

er-

a-

tion

was

suc-

cess

fully

in-

voked

get_clean

Get

the

list

of

clean

steps

from

the

ager

Parameter

task

a

Task

ager

ob-

ject

con-

tain-

ing

the

node

Raises

Node

if

the

clean

just been enrolled and has not been cleaned yet.

step
are
not
yet
avai
able
(cac
for
ex-
am-
ple,
whe
a
node
has

Returns

A
list
of
clea
step
dic-
tio-
nar-
ies

get_dep

Get
the
list
of
de-
ploy
step
from
the
ager

Param

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

has just been enrolled and has not been deployed yet.

Raises

Ins

if the deployment step are not yet available (cached) for example, when a node

Returns

A list of deployment step dictionaries

get_prop

Return the properties of the interface

class ir

Base *iron* *dri*

bas
Res
Im-
ple-
men
ta-
tion
of
Res-
cueI
ter-
face
whic
uses
ager
ram

clean_up

Clea
up
af-
ter
RES
CUE
WA
time
out/
or
fin-
ish-
ing
res-
cue.

Res-
cue
pass
wor
shou
be
re-
mov
from
the
node
and
ram
boot
en-
vi-

cleaned if Ironic is managing the ramdisk boot.

ron-
men-
shou
be

Parame

task

a
Task
ager
in-
stan
with
the
node

Raises

Net-
worl
Er-
ror
if
the
res-
cue
port
can-
not
be
re-
mov

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

rescue()

Boo
a
res-
cue
ram

on
the
node

Parameters

task
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror
if
the
ten-
ant
port
can-
not
be
re-
mov

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron

driver info is specified for power management.

Raises

othe
ex-

action.

cep-
tions
by
the
node
pow
driv
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tions

Returns

Re-
turn
state

unrescue

At-
temp
to
mov
a
res-
cue
node
back
to
ac-
tive
state

Parame

driver info is specified for power management.

task
a
Task
ager
in-
stand

Raises

Net-
world
Er-
ror
if
the
res-
cue
port
can-
not
be
re-
mov

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron

Raises

othe
ex-
cep-
tions
by
the
node

action.

pow
driv
if
som
thin
wron
oc-
curr
dur-
ing
the
pow

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tion:

Returns

Re-
turn
state

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
ager
res-
cue.

Paramete

task
a

Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
in-
stan
has
emp
pass
wor
or
res-
cu-
ing
net-

work UUID config option has an invalid value.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
mor
re-
quir
pa-

eters

ram-

class ir

Base
iro
dri
mod
age
Age
iro
dri
mod
age
Age
iro
dri
bas
Dep

A
de-
ploy
in-
ter-
face
that
re-
lies
on
a
cus-
tom
ager
to
de-
ploy
Only
pro-
vide
the
ba-
sic
de-
ploy
step
to
start
the
ram
tear

the instance boot.

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

dow
the
rame
and
pre-
pare

clean_up

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

If
im-
ple-
men
this
meth
mus

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
tear.

Paramete
task
a
Task
ager
in-
stan

deploy(A

Per-
form
a
de-
ploy
men
to
a
node

Per-
form
the
nec-
es-
sary
work
to
de-

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

ploy
an
im-
age
onto
the
spec
i-
fied
node
This

Parameter
task

a
Task
ager
in-
stan

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro

erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

Parame

task
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror:
if
the
pre-
vi-
ous
clea
ing
port
can-
not
be
re-
mov
or
if

new cleaning ports cannot be created.

driver info is specified for power management.

umes.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron

Raises
Stor
ageE
ror
If
the
stor-
age
driv
is
un-
able
to
at-
tach
the
con-
fig-
ured
vol-

Raises
othe
ex-
cep-
tions
by
the
node

action.

pow
driv
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

Raises

ex-
cep-
tion.
if
im-
age_
is
not
Glar
href
and
is
not
HTT
URI

Raises

ex-
cep-
tion.
if
net-
worl
val-
i-
da-
tion
fails

Raises

any
boot
in-
ter-
face
pre-
pare
ex-

cep-
tions

prepare_

Pre-
pare
in-
stan-
for
boot
ing.

The
base
ver-
sion
only
calls
pre-
pare
on
the
boot
in-
ter-
face

should_r

Whe
ager
boot
is
man
agec
by
iron

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.
This
meth

information for this driver to deploy images to the node.

ing.

val-
i-
date
whe
the
prop
er-
ties
of
the
sup-
plie
node
con-
tain
the
re-
quir

Paramete

task
a
Task
ager
in-
stan

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss

Raises

In-
valid

Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

ironic.d

Che
if
the
re-
ques
im-
age
is
large
than
the
ram
size.

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **image**
href
of
the
im-
age.

- **image**
The
disk
for-
mat
of
the
im-
age
if
pro-
vide

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
size
of
the
im-
age
is
grea
than
the
avai
able

ram size.

ironic.d

Val-
i-
date
con-
fig-
u-
ra-

tion
op-
tions
re-
quir
to
per-
form
HTT
pro-
vi-
sion
ing.

Paramet

nod
an
iron
node
ob-
ject

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
op-
tion
is
not
set.

`ironic.d`

Che
that
the
pro-
vide
prox
pa-
ram-
e-
ters
are
valid

incorrect.

`ironic.drivers.modules.agent_base` module

Parameter
node
an
Iron
node

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pro-
vide
prox
pa-
ram-
e-
ters
are

class ir
Base
obj

Mix
with
base
meth
ods
not
re-
ly-
ing
on
any
de-
ploy
step

clean_up

figuration files for this node.

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

Un-
link
TFT
and
in-
stan
im-
ages
and
trig-
gers
im-
age
cach
clea
Re-
mov
the
TFT
con-

Parame
task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

prepare

Boo
into
the
ager
to
pre-
pare
for
clea
ing.

Parame

task

a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Nod
Clea
ing-
Fail-
ure,
Net-
worl
Er-
ror
if
the
pre-
vi-
ous
clea
ing
port
can-
not

be removed or if new cleaning ports cannot be created.

Raises

In-
valid

invalid value.

Pa-
ram-
e-
ter-
Valu
if
clea
ing
net-
worl
UUI
con-
fig
op-
tion
has
an

Returns

state
to
sig-
nify
an
asyn
chro
pre-
pare

refresh

Re-
fres
the
node
cach
step

Paramet

- **task**
a
Task
ager
in-
stan
- **step**
clea
or

de-
ploy

should_r

When
ager
boot
is
man
aged
by
iron

take_ove

Take
over
man
age-
men
of
this
node
from
a
deac
con-
duc-
tor.

Parame

task
a
Task
ager
in-
stan

tear_dow

Tear
dow
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node

arately.

Pow
off
the
node
All
ac-
tual
clear
up
is
done
in
the
clear
meth
meth
whic
shou
be
call
sep-

Paramete

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

de-
ploy
state
DEL

Raises

Net-
worl
Er-
ror
if
the
clear
ing

port
can-
not
be
re-
mov

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
state
is
spec
i-
fied
or
the
wron
driv

info is specified.

Raises

Stor
ageE
ror
whe
vol-
ume
de-
tach
men
fails

Raises

othe
ex-
cep-
tion
by
the
node
pow
driv
if
som

action.

thing
wron
oc-
curr
dur-
ing
the
pow

tear_down

Clea
up
the
PXE
and
DHCP
files
af-
ter
clea
ing.

Parameter

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Nod
Clea
ing-
Fail-
ure,
Net-
worl
Er-
ror
if
the
clea
ing
port
can-

not
be
re-
mov

class ir

Base
iro
dri
mod
age
Hea
iro
dri
mod
age
Age

Mix
with
de-
ploy
meth
ods.

configur

Help
meth
to
con-
fig-
ure
lo-
cal
boot
on
the
node

This
meth
trig-
gers
boot
load
in-
stal-
la-
tion
on
the

tion of bootloader, this method sets the node to boot from disk.

tion which contains the image deployed or None in case of whole disk images which we expect to already have a bootloader installed.

node
On
suc-
cess-
ful
in-
stal-
la-

Parameters

- **task**
a Task object containing the node
- **root**
The UUID of the root partition. This is used for identifying the par-
- **efi**
The UUID

of
the
efi
sys-
tem
par-
ti-
tion.
This
is
used
only
in
uefi
boot
mod

- **pre**
The
UUI
of
the
PRe
Boo
par-
ti-
tion.
This
is
used
only
for
boot
ing
ppc6
hard

ware.

Raises
In-
stan-
ploy
Fail-
ure
if
boot
load
in-
stal-
la-
tion

error while setting the boot device on the node.

faile
or
on
en-
cour
ter-
ing

execute_

Ex-
e-
cute
a
clea
step
asyn
chro
on
the
ager

Parame

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **step**
a
clea
step
dic-
tio-
nary
to
ex-
e-
cute

Raises
Nod

Clea
ing-
Fail-
ure
if
the
ager
does
not
re-
turn
a
com
man
sta-
tus

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

execute

Ex-
e-
cute
a
de-
ploy
step

Wer
try-
ing
to
find
a
step
amo
both
out-
of-
band
and

plicates, out-of-band steps take priority. This property allows having an out-of-band deploy step that calls into a corresponding in-band step after some preparation (e.g. with additional input).

in-
band
step
In
case
of
du-

Parame

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **step**
a
de-
ploy
step
dic-
tio-
nary
to
ex-
e-
cute

Raises

In-
stan-
ce
ploy
Fail-
ure
if
the
ager
does
not
re-
turn

a
com
man
sta-
tus

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

get_clean

Get
the
list
of
clean
steps
from
the
agent

Parameter

tasks
a
Task
agent
ob-
ject
con-
tain-
ing
the
node

Raises

NodeError
if
the
clean
steps
are
not
yet

just been enrolled and has not been cleaned yet.

avai
able
(cac
for
ex-
am-
ple,
whe
a
node
has

Returns

A
list
of
clea
step
dic-
tio-
nar-
ies

get_dep

Get
the
list
of
de-
ploy
step
from
the
ager

Parame

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Ins
if

has just been enrolled and has not been deployed yet.

the
de-
ploy
step
are
not
yet
avai
able
(cac
for
ex-
am-
ple,
whe
a
node

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-
ies

prepare_

Pre-
pare
in-
stan-
to
boot

Paramete

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing

the
node

- **root**
the
UI
for
root
par-
ti-
tion

- **efi**
the
UI
for
the
efi
par-
ti-
tion

Raises

In-
valid
State
if
fails
to
pre-
pare
in-
stan

process

Star
the
next
clea
step
if
the
pre-
vi-
ous
one
is
com
plete
In

sion of all hardware managers at the start of the process (the agents `get_clean|deploy_steps()` call) and before executing each step. If the version has changed between steps, the agent is unable to tell if an ordering change will cause an issue so it returns `VERSION_MISMATCH`. For automated cleaning, we restart the entire cleaning cycle. For manual cleaning or deploy, we dont.

coordinate the reboot once the step is completed.

or-
der
to
avoi
er-
rors
and
mak
agen
up-
grad
pain
less,
the
agen
com
pare
the
ver-

Ad-
di-
tion-
ally,
if
a
step
in-
clud
the
re-
boot
prop
erty
set
to
True
this
meth
will

reboot_a

Help
meth
to
trig-

complete. On failure, it logs the error and marks deploy as failure.

ger
re-
boot
on
the
node
and
fin-
ish
de-
ploy

This
meth
ini-
ti-
ates
a
re-
boot
on
the
node
On
suc-
cess
it
mark
the
de-
ploy
as

Parame

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

In-
stan
ploy
Fail-
ure,

if
node
re-
boot
faile

refresh

Re-
fresh
the
node
cach
clea
step
from
the
boot
ager

Gets
the
node
step
from
the
boot
ager
and
cach
then
The
step
are
cach
to
mak
get_
calls
syn-

chronous, and should be refreshed as soon as the agent boots to start cleaning/deploy or if cleaning is restarted because of a hardware manager version mismatch.

Parame

- **task**
a
Task
ager
in-
stan

sults

- **step**
clear
or
de-
ploy

Raises

Node
Clearing-
ing-
Failure
ure
or
In-
stan-
ploy
Fail-
ure
if
the
agen-
re-
turn
in-
valid
re-

tear_down

A
de-
ploy
step
to
tear
down
the
agen-

Parameter

task
a
Task
agen-
ob-
ject
con-
tain-
ing
the
node

class ir
Base
obj
Mix
with
out-
of-
band
de-
ploy
step

boot_in
De-
ploy
step
to
boot
the
fi-
nal
in-
stan

Parame
task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

switch_t
De-
ploy
step
to
swit
the
node
to
the
ten-
ant
net-

world

Parameter

task

a

Task

ager

ob-

ject

con-

tain-

ing

the

node

class ir

Base

obj

Mix

class

im-

ple-

men

ing

hear

beat

pro-

cess

ing.

continua

Star

the

next

clear

ing

step

if

the

pre-

vi-

ous

one

is

com

plete

Parameter

task

a

Task

ager
in-
stan

heartbe

Pro-
cess
a
hear
beat

Parame

- **task**
task
to
worl
with
- **cal**
ager
HTT
API
URI
- **ager**
The
ver-
sion
of
the
ager
that
is
hear
beat
ing
- **ager**
TLS
cer-
tifi-
cate
for
the
ager
- **ager**

Sta-
tus
of
the
hear
beat
ing
ager

- **ager**
Sta-
tus
mes
sage
that
de-
scrib
the
ager

property
De-
fine
node
state
when
hear
beat
ing
is
al-
lowe

process
Star
the
next
clear
step
if
the
pre-
vi-
ous
one
is
com
plete

Paramete

- **task**
a
Task
ager
in-
stan

- **step**
clea
or
de-
ploy

reboot_t

Met
in-
voke
af-
ter
the
de-
ploy
men
is
com
plete

Parame

task
a
Task
ager
in-
stan

refresh

Re-
fresh
the
node
cach
clea
step

Parame

task
a
Task
ager
in-

stan

refresh

Re-
fresh
the
node
cach
clea
step

Paramet

- **task**
a
Task
ager
in-
stan

- **step**
clea
or
de-
ploy

ironic.d

ironic.d

Ex-
e-
cute
a
clea
or
de-
ploy
step
asyn
chro
on
the
ager

Paramet

- **task**
a

Task
ager
ob-
ject
con-
tain-
ing
the
node

- **step**
a
step
dic-
tio-
nary
to
ex-
e-
cute

- **step**
clea
or
de-
ploy

- **cli**
ager
clien
(if
avai
able

Raises

Nod
Clea
ing-
Fail-
ure
(clea
step
or
In-
stan
ploy
Fail-
ure
(de-
ploy

does not return a command status.

step
if
the
ager

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

`ironic.d`

Find
the
give
in-
band
step

`ironic.d`

Get
the
list
of
cach
clear
or
de-
ploy
step
from
the
ager

The
step
cach
is
up-
date
at
the
be-

gin-
ning
of
clea
ing
or
de-
ploy

Paramet

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **step**
clea
or
de-
ploy
- **int**
The
in-
ter-
face
for
whic
clea
step
are
to
be
re-
turn
If
this
is
not
pro-

vided, it returns the steps for all interfaces.

ities for them. If a step isnt in this dictionary, the steps original priority is used.

- **over:**
a
dic-
tio-
nary
with
keys
be-
ing
step
nam
and
val-
ues
be-
ing
new
pri-
or-

Returns

A
list
of
clea
step
dic-
tio-
nar-
ies

ironic.d

Help
meth
to
log
the
er-
ror
and
raise
ex-
cep-
tion.

Paramet

-

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **msg**
the
mes-
sage
to
set
in
last_
of
the
node

- **col**
Boo-
in-
di-
cat-
ing
whe-
to
at-
temp
to
col-
lect
logs
from
IPA-
base
ram-
De-

faults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

- **exc**
Ex-

cep-
tion
that
caus
the
fail-
ure.

ironic.d

Dec
o-
ra-
tor
meth
for
addi
a
post
clea
step
hook

This
is
a
mec
a-
nism
for
addi
a
post
clea
step
hook
for
a
par-
tic-
u-
lar
clea

step. The hook will get executed after the clean step gets executed successfully. The hook is not invoked on failure of the clean step.

Any
meth
to
be
mad
as
a

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the clean step and the node will be moved to CLEANFAIL state.

Parameters

- **interface**
The name of the interface after which it should be executed.
- **step**
The name of the step after which it should be executed.

Returns

A method which registers is-

ters
the
give
meth
as
a
post
clea
step
hool

ironic.d

Dec
o-
ra-
tor
meth
for
addi
a
post
de-
ploy
step
hool

This
is
a
mec
a-
nism
for
addi
a
post
de-
ploy
step
hool
for
a
par-
tic-
u-
lar

deploy step. The hook will get executed after the deploy step gets executed successfully. The hook is not invoked on failure of the deploy step.

Any
meth
to

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the deploy step and the node will be moved to DEPLOYFAIL state.

be
mad
as
a
hook
may
be
dec-
o-
rate
with
@pc
men
tion-
ing
the
in-

Parameter

- **interface**
The name of the interface.
- **step**
The name of the step after which it should be executed.

Returns

A

meth
whic
reg-
is-
ters
the
give
meth
as
a
post
de-
ploy
step
hook

`ironic.drivers.modules.agent_client` module

class `ir`
Base
obj

Clie
for
in-
ter-
act-
ing
with
node
via
a
RES
API

collect
Col-
lect
and
pack
age
di-
ag-
nos-
tic
and
sup-
port
data
from

the
rame

Parameter
node

A
Node
ob-
ject.

Raises

Iron
icEx
cep-
tion
when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

sponse from the agent.

Raises

Age
tAPI
ror
when
agent
failed
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
when

ing the prior command.

sample.

the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
[get](#).
for
a
com
man
re-
sult

execute_

Ex-
e-
cute
spec
i-
fied
clear
step

Parame

- **step**
A
clea
step
dic-
tio-
nary
to
ex-
e-
cute

- **node**
A
Node
ob-
ject.

- **port**
Port
as-
so-
ci-
ated
with
the
node

Raises
Iron
icEx
cep-
tion
when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

sponse from the agent.

ing the prior command.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

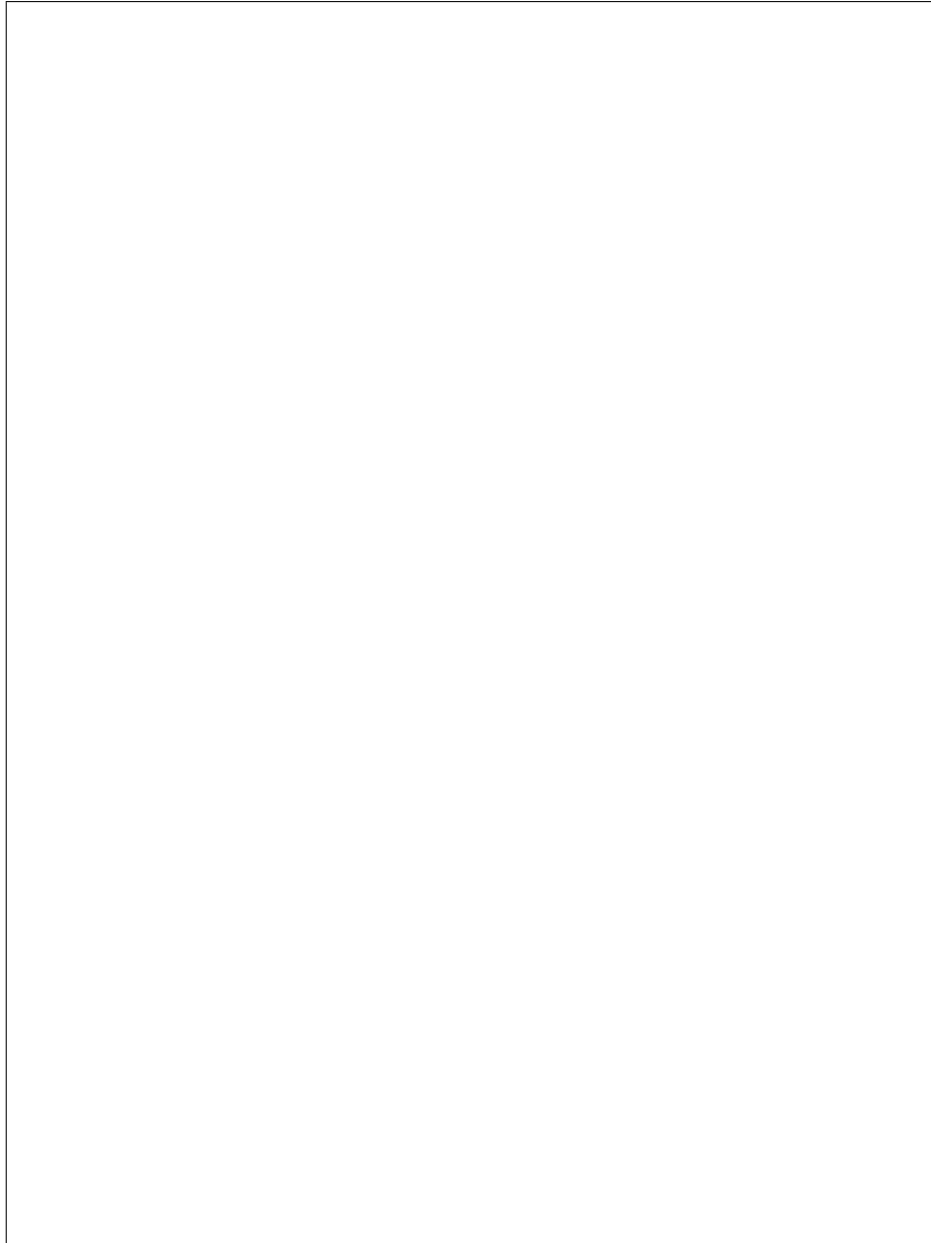
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

See
get.
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



execute_

Ex-
e-
cute
spec

i-
fied
de-
ploy
step

Parame

- **step**
A
de-
ploy
step
dic-
tio-
nary
to
ex-
e-
cute

- **nod**
A
Nod
ob-
ject.

- **por**
Port
as-
so-
ci-
ated
with
the
node

Raises

Iron
icEx
cep-
tion
when
faile
to
is-
sue
the
re-
ques

sponse from the agent.

ing the prior command.

or
there
was
a
mal-
form
re-

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

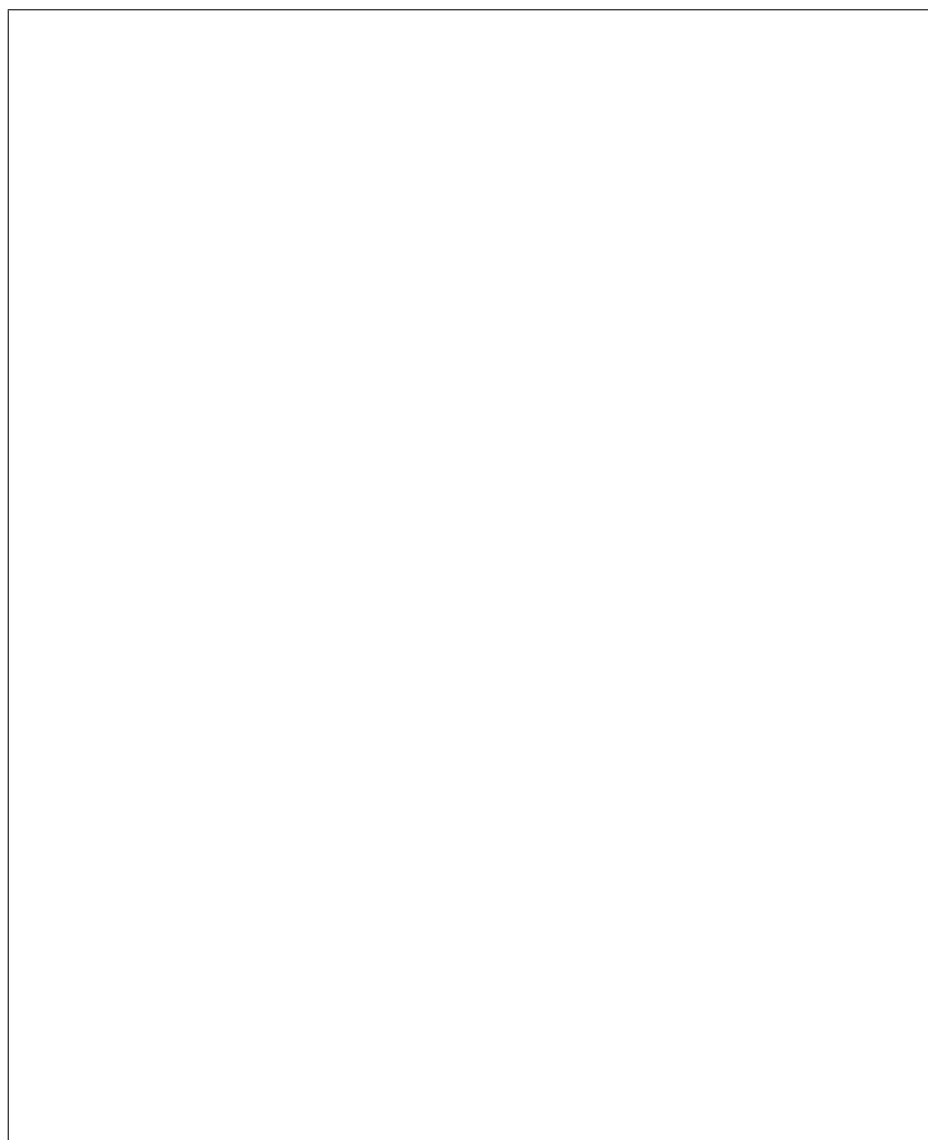
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict

con-
tain-
ing
com
man
re-
spor
from
ager
See
get.
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



finalize

In-
struc-
the
ram-
to
fi-
nal-
ize
en-
ter-
ing
of
res-
cue
mod

Paramete

node
A
Nod
ob-
ject.

Raises

Iron
icEx
cep-
tion
if
res-
cue
is
miss
ing,
or
whe
faile
to
is-
sue
the
re-
ques

or there was a malformed response from the agent.

Raises

Age

ing the prior command.

tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Raises

In-
stan
cue-
Fail-
ure
whe
the
ager
ram
is
too
old
to
sup-

the rescue password.

sample.

port
trans
mis-
sion
of

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get.
for
a
com
man
re-
sult

get_clea

Get
clea
step
from
ager

Parame

- **node**
A
node
ob-
ject.
- **por**
Port
as-
so-
ci-
ated

sponse from the agent.

with
the
node
Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

Raises
Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises
Age
Prog
whe
the
com
man
fails
to
ex-

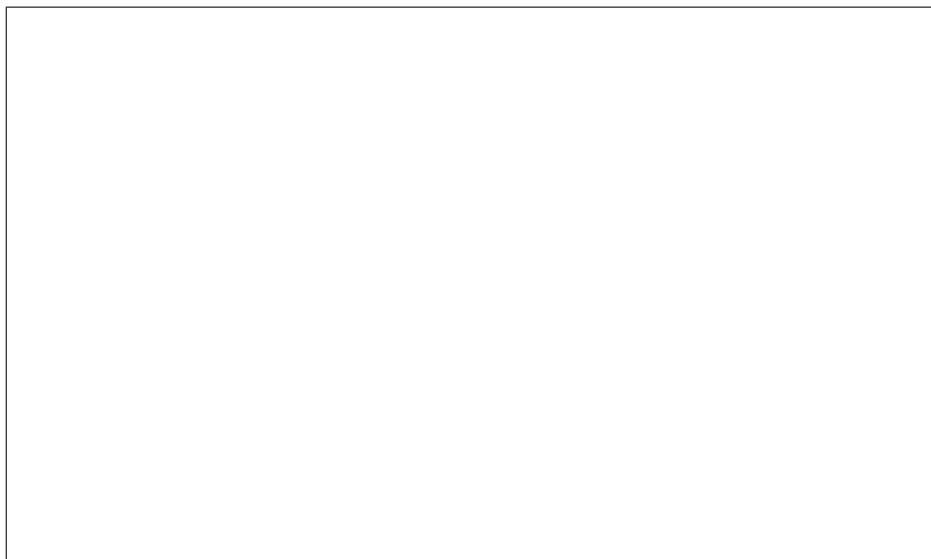
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

ing the prior command.

Returns

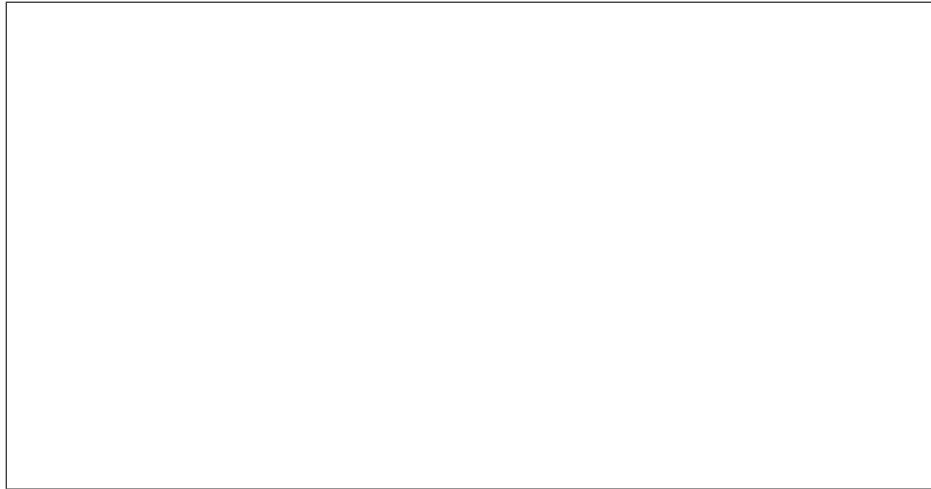
A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get.
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



get_com

Get
com
man
sta-
tus
from
ager

Parame

- **nod**
A
Nod
ob-
ject.
- **ret:**
Whe
to
retry
con-
nec-
tion
prob
lems
- **exp**
If
True
do
not
log

con-
nec-
tion
prob
lems
as
er-
rors.

Returns

A
list
of
com
man
re-
sults
each
re-
sult
is
re-
lated
to
a
com
man
been
is-
sued

to agent. A typical result can be:



(continues on next page)

(continued from previous page)



↔e.g. 'RUNNING', 'SUCCEEDED', 'FAILED'>

(continues on next page)

(continued from previous page)



↪succeeded, the value **is** command specific,

(continues on next page)

(continued from previous page)

↔* a dictionary containing keys `clean_result`

↔ `and` `clean_step` `for` the command

(continues on next page)

(continued from previous page)

↔* a dictionary containing keys `deploy_result`

↔ `and` `deploy_step` `for` the command

(continues on next page)

(continued from previous page)

↔* a string representing result message **for**

↔ the command `standby.cache_image;`

(continues on next page)

(continued from previous page)



get_dep:

Get
de-
ploy
step
from
agen

Parame

- **node**
A
node
ob-
ject.

- **port**
Port
as-
so-
ci-
ated
with
the
node

Raises

Iron
icEx
cep-
tion
when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form

sponse from the agent.

re-

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

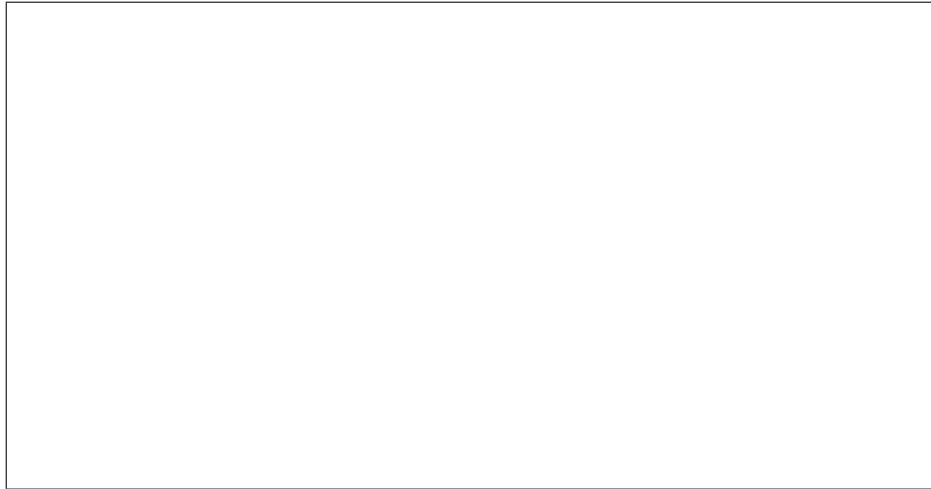
A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get.
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



get_last

Get the last status for the given command

Parameters

- **node**
A Node object.
- **method**
Command name

Returns

A dictionary containing command status

found.

sponse from the agent.

tus
from
ager
or
Non
if
the
com
man
was
not

get_part

Get
de-
ploy
step
from
ager

Parame

node
A
node
ob-
ject.

Raises

Iron
icEx
cep-
tion
when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

Raises

Age

tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

ing the prior command.

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

install

In-

stall
a
boot
load
on
the
im-
age.

Parame

- **node**
A
node
ob-
ject.
- **root**
The
UI
of
the
root
par-
ti-
tion.
- **target**
The
tar-
get
de-
ploy-
ment
boot
mod
- **efi**
The
UI
of
the
efi
sys-
tem
par-
ti-
tion
whe

to, only used for uefi boot mode.

when local booting a partition image on a ppc64* system.

sponse from the agent.

the
boot
load
will
be
in-
stall

- **pre**
The
UI
of
the
PRe
Boo
par-
ti-
tion
whe
the
boot
load
will
be
in-
stall
to

Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

ing the prior command.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See

sample.

get.
for
a
com
man
re-
sult

power_o:

Soft
pow
ers
off
the
bare
meta
node
by
shut
ting
dow
ram
OS.

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

sponse from the agent.

ing the prior command.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

sample.

See
[get](#).
for
a
com
man
re-
sult

reboot (r)

Soft
re-
boot
the
bare
meta
node
by
shut
ting
dow
ram
OS.

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

sponse from the agent.

ing the prior command.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

sample.

sponse from the agent.

See
[get](#).
for
a
com
man
re-
sult

sync(*node*)

Flush
file
sys-
tem
buff
forc-
ing
char
bloc
to
disk

Parameter

node
A
Node
ob-
ject.

Raises

Iron
icEx
cep-
tion
when
fail
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-

ing the prior command.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the
ager
is
pres
ex-
e-
cut-

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See

sample.

get.
for
a
com
man
re-
sult

ironic.d

Ex-
tract
an
er-
ror
strin
from
the
com
man
re-
sult.

Paramet

com
Com
man
in-
for-
ma-
tion
from
the
ager

Returns

Er-
ror
strin

ironic.drivers.modules.agent_power module

The
ager
pow
in-
ter-
face

class ir

Base
iron
driver
base
Power
Pow
in-
ter-
face
us-
ing
the
run-
ning
agen
for
pow
ac-
tions

get_power

Re-
turn
the
pow
state
of
the
task
node
Es-
sen-
tially,
the
only
know
state
is
POW
ON,
ev-
ery-
thing
else
is
an
er-
ror

precisely None).

(or
more

Parameters

task

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Returns

A

pow

state

One

of

iron

com

sta

get_prop

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip
tion:
en-
tries

get_supp

Get
a
list
of
the
sup-
port
pow
state

Only
con-
tains
RE-
BOC

Paramete

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
pow
state
de-
finec
in
iron
com

sta

reboot (i

Per-
form
a
re-
boot
of
the
task
node
Only
soft
re-
boot
is
im-
ple-
men

Paramete

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-

indicates to use default timeout.

ger
(>
0)
for
any
pow
state
Non

set_power

Set
the
pow
state
of
the
task
node

Parameter

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **power**
Pow
state
from
ironiC
command
state
Only
RE-
BOC
and
SOF

mous.

indicates to use default timeout.

are
sup-
port
and
are
syn-
ony-

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

Raises
Pow
er-
State
Fail-
ure
on
non-
supp
pow
state

support:
Che
if
pow
sync
is
sup-
port
for

the
give
node

Not
sup-
port
for
the
ager
pow
sinc
it
is
not
pos-
si-
ble
to
pow
on/o
node

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on
with
a
shar
lock

Returns

bool
whe
pow
sync
is
sup-
port

validat

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

**Parame
task**

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

ironic.drivers.modules.boot_mode_utils module

ironic.d

Con-
fig-
ures
se-
cure
boot
if
it
has
been
re-
ques
for
the
node

ironic.d

De-
con-
fig-
ures
se-
cure
boot
if
it
has
been
re-
ques
for
the
node

ironic.d

Re-
turn
the
boot
mod

Parameter

node
an
iron
node
ob-

to node properties/capabilities

ject.
Returns
bios
or
uefi
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
node
boot
mod
dis-
agre
with
the
boot
mod
set

ironic.d

Re-
turn
the
boot
mod
that
wou
be
used
for
de-
ploy
This
meth
re-
turn
boot
mod
to
be
used

set to true or returns bios if trusted_boot is set to true in instance_info/capabilities of node. Otherwise it returns value of boot_mode in properties/capabilities of node if set. If that is not set, it returns boot mode in internal_driver_info/deploy_boot_mode for the node. If that is not set, it returns boot mode in instance_info/deploy_boot_mode for the node. It would return None if boot mode is present neither in capabilities of node properties nor in nodes internal_driver_info nor in nodes instance_info (which could also be None).

to node properties/capabilities

for
de-
ploy
It
re-
turn
uefi
if
se-
cure
is

Parameter
node

an
ironic
node
ob-
ject.

Returns

bios
uefi
or
None

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
node
boot
mod
dis-
agre
with
the
boot
mod
set

ironic.d

Re-
turn
True
if
se-
cure
is
re-
ques
for
de-
ploy

This
meth
check
node
prop
erty
for
se-
cure
and
re-
turn
True
if
it
is
re-
ques

Paramet

nod
a
sin-
gle
Nod

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-

tionary or is malformed.

bil-
i-
ties
strin
is
not
a
dic-

Returns

True
if
se-
cure
is
re-
ques

`ironic.d`

Re-
turn
True
if
trust
is
re-
ques
for
de-
ploy

This
meth
chec
in-
stan
prop
erty
for
trust
and
re-
turn
True
if
it
is
re-
ques

Paramet

tionary or is malformed.

node
a
single
Node

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-
ties
strin
is
not
a
dic-

Returns

True
if
trust
is
re-
ques

ironic.d

Set
node
boot
mod
from
bare
meta
con-
fig-
u-
ra-
tion
At-
temp

configuration:

set and apply the logic that follows

to
read
cur-
rentl
set
boot
mod
off
the
bare
meta
ma-
chin
Also
read
node
boot
mod

- If BM drive does not implement getting boot mod as sum BM boot mod is not

- If Iron node boot mod is not set and

Ironic boot mode to *[deploy]/default_boot_mode*

node boot mode on the Ironic node

BM
node
boot
mod
is
not
set
-
set

- If
Iron
node
boot
mod
is
not
set
and
BM
node
boot
mod
is
set
-
set
BM

- If
Iron
node
boot
mod
is
set
and
BM
node
boot
mod
is
not
set
-

boot mode to BM boot mode

Ironic boot mode to BM boot mode and fail hard if underlying hardware type does not support setting boot mode

set
Iron

- If both Ironic and BM nodes boot mode are set but they differ -

try
to
set

In the end, the new boot mode may be set in drive

Parameter
task
a
task
ob-
ject

ironic.drivers.modules.console_utils module

Iron
con-
sole
util-
i-
ties.

ironic.d

Re-
turn
a
free
TCP
port
on
cur-
rent
host

Find
and
re-
turn
a
free
TCP
port
in
the
rang
of
CON

ironic.d

Get
a
url
to
ac-
cess
the
con-
sole
via
shel
linal

Paramet

por
the
ter-
mi-
nal
port
for
the
node

`ironic.d`

Get
a
URI
to
ac-
cess
the
con-
sole
via
so-
cat.

Paramet

por
the
ter-
mi-
nal
port
(in-
te-
ger)
for
the
node

Returns

an
ac-
cess
URI
to
the
so-
cat
con-
sole
of
the
node

ironic.d

Writ
a
file
con-
tain-
ing
a
pass
wor
un-
til
dele

ironic.d

Re-
lease
spec
i-
fied
TCP
port

ironic.d

Ope
the
se-
rial
con-
sole
for
a
node

Paramet

- **node**
the
uuid
for
the
node
- **port**
the
ter-
mi-
nal

port
for
the
node

- **con.**
the
shel
com
man
that
gets
the
con-
sole

Raises

Con
sole
ror
if
the
di-
rec-
tory
for
the
PID
file
can-
not
be
cre-
ated
or
an

old process cannot be stopped.

Raises

Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing
the
sub-
pro-

cess
faile

ironic.d

Ope
the
se-
rial
con-
sole
for
a
node

Paramet

- **node**
the
uuid
of
the
node
- **port**
the
ter-
mi-
nal
port
for
the
node
- **con**
the
shel
com
man
that
will
be
ex-
e-
cute
by
so-
cat
to
es-

sole to the node

tab-
lish
con-

Raises

- **Con.**
if
the
di-
rec-
tory
for
the
PID
file
or
the
PID
file
can-
not
be
cre-
ated
- **Con.**
whe
in-
vok-
ing
the
sub-
pro-
cess
faile

ironic.d

Clos
the
se-
rial
con-
sole
for
a
node

Paramet

node
the
UI
of
the
node

Raises

Con
sole
ror
if
un-
able
to
stop
the
con-
sole
pro-
cess

ironic.d

Clos
the
se-
rial
con-
sole
for
a
node

Paramet

node
the
UI
of
the
node

Raises

Con
if
un-
able
to
stop
the
con-
sole
pro-

ironic.drivers.modules.deploy_utils module

tion to IPA.

class `ironic.drivers.modules.deploy_utils`
Base class for the `ironic.drivers.modules.deploy_utils` module.
`ironic.drivers.modules.deploy_utils`
`ironic.drivers.modules.deploy_utils`
`ironic.drivers.modules.deploy_utils`
`ironic.drivers.modules.deploy_utils`

`ironic.drivers.modules.deploy_utils`
Add the required configuration parameters to the node driver.
Add the required configuration options to the node driver.
It is required to pass the information to the node driver.

Parameter
`task`
a

Task
ager
in-
stan

ironic.d

Buil
the
op-
tion
to
be
pass
to
the
ager
ram

Paramet

nod
an
iron
node
ob-
ject

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
pa-
ram-
e-
ters
to
be
pass
to
ager
ram

ironic.d

Buil
in-
stan
nec-
es-

sary
for
de-
ploy
ing
to
a
node

Parameter

task
a
Task
manager
object
containing
the
node

Returns

a
dictionary
containing
the
properties
to
be
updated
in
instance

Raises

exception.
if
image_
is
not
Glan
href

and
is
not
HTT
URI

ironic.d

Fetc
the
in-
stan
im-
age
from
Glar

This
meth
pulls
the
disk
im-
age
and
writ
then
to
the
ap-
pro-
pri-
ate
plac
on
lo-
cal

disk.

Paramet

- **ctx**
con-
text
- **node**
an
iron
node
ob-
ject

- **for**
when
con-
vert
im-
age
to
raw
for-
mat

Returns

a
tu-
ple
con-
tain-
ing
the
uuid
of
the
im-
age
and
the
path
in
the
files
tem

where image is cached.

`ironic.d`

Che
for
emp
para
in
the
pro-
vide
dic-
tio-
nary

Paramet

- **inf**

missing parameters.

The
dic-
tio-
nary
to
in-
spec

- **err**
The
er-
ror
mes-
sage
to
pre-
fix
be-
fore
prin-
ing
the
in-
for-
ma-
tion
about

- **par**
Add
this
pre-
fix
to
each
pa-
ram-
e-
ter
for
er-
ror
mes-
sage

Raises
Miss-
ing-
Pa-
ram-

provided dictionary.

e-
ter-
Valu
if
one
or
mor
pa-
ram-
e-
ters
are
emp
in
the

ironic.d

Eval
u-
ate
in-
ter-
face
to
de-
ter-
mine
if
ca-
pa-
bil-
ity
is
pres

Paramet

- **int**
The
in-
ter-
face
ob-
ject
to
chec
- **cap**

present.

The
valu
rep-
re-
sent
ing
the
ca-
pa-
bil-
ity
that
the
calle
wish
to
chec
if

Returns

True
if
ca-
pa-
bil-
ity
foun
oth-
er-
wise
Fals

ironic.d

Com
pute
chec
sum
by
give
im-
age
path
and
al-
go-
rithr

ironic.d

Dele
in-

stand
im-
age
file
and
sym
boli
link
refer
to
it.

ironic.d

Dele
in-
stan
im-
age
file.

Paramet

nod
the
uuid
of
the
iron
node

ironic.d

Whe
con-
vert
im-
age
to
raw
for-
mat
for
spec
i-
fied
node

Paramet

nod
iron
node
ob-
ject

Returns

Boolean value indicating whether the direct deployment interface should convert image to raw.

ironic.d

Check for available disk space and fetch images using ImageCache.

Parameters

- **ctx**: context
- **cache**: ImageCache instance to

use
for
fetch
ing

- **image**
list
of
tu-
ples
(im-
age
href
des-
ti-
na-
tion
path

- **format**
bool
valu
whe
to
con-
vert
the
im-
age
to
raw
for-
mat

Raises
In-
stan-
ce
ploy
Fail-
ure
if
un-
able
to
find
enou
disk
spac

ironic.d

Re-
turn
state
base
on
op-
er-
a-
tion
(clea
ing/
be-
ing
in-
voke

Parameter
node
an
ironic
node
ob-
ject.

Returns
state
if
clea
ing
op-
er-
a-
tion
in
prog
or
state
if
de-
ploy
op-
er-
a-
tion

in progress.

ironic.d

Gets
the
boot
op-

or is malformed.

tion.
Parameter
node
A
sin-
gle
Nod

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-
ties
strin
is
not
a
dict

Returns
A
strin
rep-
re-
sent
ing
the
boot
op-
tion
type
De-
fault
to
net-
boot

ironic.d
Gets
the

de-
fault
boot
op-
tion.

ironic.d

Re-
turn
the
disk
la-
bel
re-
ques
for
de-
ploy
if
any.

Paramet

nod

a
sin-
gle
Nod

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-
ties
strin
is
not
a
dic-

tionary or is malformed.

Returns

the

disk
la-
bel
or
Non
if
no
disk
la-
bel
was
spec
i-
fied.

ironic.d

Get
the
ef-
fec-
tive
valu
of
im-
age_
for
the
node

ironic.d

Gets
the
im-
age
in-
for-
ma-
tion
from
the
node

Get
im-
age
in-
for-
ma-
tion
for
the

give
node
in-
stan-
from
its
in-
stan-
prop
erty.

Parameter
node

a
sin-
gle
Node

Returns

A
dict
with
re-
quir
im-
age
prop
er-
ties
re-
triev
from
node
in-
stan-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
im-
age_
is
miss
ing
in
node
in-

same exception if kernel/ramdisk is missing in instance_info for non-glance images.

boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

stand
Also
raise

ironic.d

Re-
turn
the
iPXE
boot
file
nam
re-
ques
for
de-
ploy

This
meth
re-
turn
iPXE
boot
file
nam
to
be
used
for
de-
ploy
Ar-
chi-
tec-
ture
spe-
cific

If
no
valid
valu
is
foun
the
de-
fault
re-
vert.

[pxe]uefi_ipxe_bootfile_name settings.

to
the
get.
meth
and
thus
the
[px
and

**Paramet
nod**

A
sin-
gle
Nod

Returns

The
iPX
boot
file
nam

ironic.d

Re-
turn
the
iPX
con-
fig
tem-
plate
file
nam
re-
ques
of
de-
ploy

This
meth
re-
turn
the
iPX
con-
fig-
u-
ra-

tion
tem-
plate
file.

Parameter
node

A
sin-
gle
Node

Returns

The
iPXE
con-
fig
tem-
plate
file
nam

`ironic.d`

Re-
solv
Iron
API
end-
poin

ei-
ther
from
con-
fig
of
from
Key
ston
cat-
a-
log.

`ironic.d`

Re-
turn
the
PXE
boot
file
nam
re-

boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

ques
for
de-
ploy

This
meth
re-
turn
PXE
boot
file
nam
to
be
used
for
de-
ploy
Ar-
chi-
tec-
ture
spe-
cific

Parameter
node

A
sin-
gle
Node

Returns

The
PXE
boot
file
name

ironic.d

Re-
turn
the
PXE
con-
fig
tem-
plate
file
name

template is searched in the node. After that architecture specific template file is searched. BIOS/UEFI template file is used if no valid architecture specific file found.

re-
ques
for
de-
ploy
This
meth
re-
turn
PXE
con-
fig
tem-
plate
file
to
be
used
for
de-
ploy
First
spe-
cific
pxe

Paramet
nod
A
sin-
gle
Nod

Returns
The
PXE
con-
fig
tem-
plate
file
nam

ironic.d
Iden
tify
a
boot
vol-

ume
from
any
con-
fig-
ured
vol-
ume

Returns

Non
or
the
vol-
ume
tar-
get
rep-
re-
sent
ing
the
vol-
ume

ironic.d

Get
a
root
de-
vice
re-
ques
for
de-
ploy
men
or
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
hint

Returns

Pars
root
de-
vice
hint
or
Non
if
no
hint
were
pro-
vide

ironic.d

Re-
turn
the
MA
ad-
dres
of
a
port
whic
has
a
VIF
port
id.

Paramet

task
a
Task
ager
in-
stan
con-
tain-
ing
the
port
to
act
on.

Returns

MA
ad-
dres

find any port with vif id.

of
the
port
con-
nect
to
de-
ploy
men
net-
worl
Non
if
it
can-
not

ironic.d

De-
ter-
min
if
Ana
con
de-
ploy
in-
ter-
face
is
in
use
for
the
de-
ploy
men

Paramet

node

A
sin-
gle
Nod

Returns

A
bool
valu
of
True

False

when
Ana
con
de-
ploy
in-
ter-
face
is
in
use
oth-
er-
wise

ironic.d

Re-
turn
true
if
boot
ing
from
an
iscsi
vol-
ume

ironic.d

ironic.d

De-
ter-
min
if
soft-
ware
raid
is
in
use
for
the
de-
ploy
men

Paramet
nod
A

sin-
gle
Nod

Returns

A
bool
valu
of
True
whe
soft-
ware
raid
is
in
use,
oth-
er-
wise
Fals

ironic.d

Gets
the
in-
stan
spe-
cific
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
in-
stan
prop
erty
of
the
sup-
plie
node
con-

quired information for this driver to deploy images to the node.

ing.

tains
the
re-

Parameter
node
a
sin-
gle
Nod

Returns
A
dict
with
the
in-
stan-
val-
ues.

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
any
of
the
re-
quir-
pa-
ram-
e-
ters
are
miss-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu-
if

any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

ironic.d

Set
node
drive
for
boot
from
vol-
ume
pa-
ram-
e-
ters.

Paramet

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Stor
ageE
ror
whe
a
node
has
an
iSC
or
Fi-
breC

is not capable to support it.

updates the clean parameters in nodes driver_internal_info. 2. If manage_boot parameter is set to true, it also calls the prepare_ramdisk method of boot interface to boot the agent ramdisk. 3. Reboots the bare metal node.

nel
boot
vol-
ume
de-
fine
but

ironic.d

Pre-
pare
the
node
to
boot
into
agen
for
in-
band
clean
ing.

This
meth
does
the
fol-
low-
ing:
1.
Pre-
pare
the
clean
ing
port
for
the
bare
meta
node
and

Paramet

•

to boot the agent ramdisk. If False, it skips preparing the boot agent ramdisk using boot interface, and assumes that the environment is setup to automatically boot agent ramdisk every time bare metal node is rebooted.

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

- **man**
If
this
is
set
to
True
this
meth
calls
the
pre-
pare
meth
of
boot
in-
ter-
face

Returns
state
to
sig-
nify
an
asyn
chro
pre-
pare

Raises
Net-
work
Er-
ror,
Nod

be removed or if new cleaning ports cannot be created.

invalid value.

Clea
ing-
Fail-
ure
if
the
pre-
vi-
ous
clea
ing
port
can-
not

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
clea
ing
net-
worl
UI
con-
fig
op-
tion
has
an

ironic.d

ironic.d

ironic.d

Sets
ap-
pro-
pri-
ate
re-
boot

flags
in
drive
base
on
op-
er-
a-
tion

Parameter

- **node**
an
iron
node
ob-
ject.

- **reboot**
Boo
valu
to
set
for
node
drive
flag
clear
ing_
or
de-
ploy
men
base
on
clear
ing

or deployment operation in progress. If it is None, corresponding reboot flag is not set in nodes driver_internal_info.

- **skip**
Boo
valu
to
set
for
node
drive

ment operation in progress. If it is None, corresponding skip step flag is not set in nodes driver_internal_info.

the corresponding polling flag is not set in the nodes driver_internal_info.

flag
skip
or
skip
base
on
clear
ing
or
de-
ploy

- **polling**
Boolean
value
to
set
for
node
driver
flag
de-
ploy
men
or
clear
ing_
If
it
is
Non

ironic.d

Sets
the
de-
ploy
sta-
tus
as
failed
with
rel-
e-
vant
mes-
sage

to DEPLOYFAIL and updates last_error with the given error message. It also powers off the baremetal node.

This
meth
sets
the
de-
ploy
men
as
fail
with
the
give
mes
sage
It
sets
node
pro-
vi-
sion

Parameter

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **msg**
the
mes
sage
to
set
in
logs
and
last_

faults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

of
the
node

- **col**
Boo
in-
di-
cat-
ing
whe
to
at-
temp
to
col-
lect
logs
from
IPA-
base
ram
De-

ironic.d

Swit
a
pxe
con-
fig
from
de-
ploy
men
mod
to
ser-
vice
mod

Paramet

-

age.

patl
path
to
the
pxe
con-
fig
file
in
tftp-
boot

- **root**
root
uuid
in
case
of
par-
ti-
tion
im-
age
or
disk
in
case
of
who
disk
im-

- **boot**
if
boot
mod
is
uefi
or
bios

- **is_1**
if
the
im-
age
is
a
who

clusive. You can have one or neither, but not both.

disk
im-
age
or
not.

- **tru:**
if
boot
with
trust
or
not.
The
us-
age
of
is_w
and
trust
are
mu-
tu-
ally
ex-

- **isc:**
if
boot
is
from
an
iSCS
vol-
ume
or
not.

- **ram:**
if
the
boot
is
to
be
to
a
ram
con-

fig-
u-
ra-
tion.

- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the
calle
is
us-
ing
iPX

- **ana**
if
the
boot
is
to
be
to
an
ana-
conc
con-
fig-
u-
ra-
tion.

ironic.d

Tear
dow
the
en-
vi-
ron-
men
setu

fast tracked or there was a cleaning failure). 2. If `manage_boot` parameter is set to true, it also calls the `clean_up_ramdisk` method of boot interface to clean up the environment that was set for booting agent ramdisk. 3. Deletes the cleaning ports which were setup as part of cleaning.

for
in-
band
clean
ing.
This
meth
does
the
fol-
low-
ing:
1.
Pow
ers
off
the
bare
meta
node
(un-
less
the
node
is

Parameter

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **manage_boot**
If
this
is
set
to

boot the agent ramdisk. If False, it skips this step.

True
this
meth
calls
the
clea
meth
of
boot
in-
ter-
face
to

Raises

Net-
worl
Er-
ror,
Nod
Clea
ing-
Fail-
ure
if
the
clea
ing
port
can-
not
be
re-
mov

ironic.d

Clea
up
stor-
age
con-
fig-
u-
ra-
tion.
Re-
mov
en-
tries
from

drive
for
stor-
age
and
dele
the
vol-
ume
tar-
gets
from
the
data
This

is done to ensure a clean state for the next boot of the machine.

ironic.d

Trie
to
set
the
boot
de-
vice
on
the
node

This
meth
tries
to
set
the
boot
de-
vice
on
the
node
to
the
give
boot
de-
vice
Un-
der

uefi boot mode, setting of boot device may differ between different machines. IPMI does not work for setting boot devices in uefi mode for certain machines. This method ignores the expected IPMI failure

for uefi boot mode and just logs a message. In error cases, it is expected the operator has to manually set the node to boot from the correct device.

Parameter

- **task**
a Task manager object containing the node
- **device**
the boot device
- **persistent**
Whether to set the boot device persistently

Raises

Any exception from set_
except IP-MI-Failure (setting

ing ipmi is expected to fail).

bilities. For all supported capabilities specified for a Node, it validates that it has a valid value. The node can have capability as part of the properties or instance_info or both. Note that the actual value of a capability does not need to be the same in the nodes properties and instance_info.

of
boot
de-
vice
us-

ironic.d

Val-
i-
date
that
spec
i-
fied
sup-
port
ca-
pa-
bil-
i-
ties
have
valid
valu

This
meth
chec
if
the
any
of
the
sup-
port
ca-
pa-
bil-
ity
is
pres
in
Nod
ca-
pa-

**Paramet
nod**

value.

an
iron
node
ob-
ject.
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
ity
is
not
set
to
a
valid

ironic.d

Val-
i-
date
the
im-
age.
For
Glar
im-
ages
it
chec
that
the
im-
age
ex-
ists
in
Glar
and
its

deployment info contain the properties passed. If its not a Glance image, it checks that deployment info contains needed properties.

prop
er-
ties
or

Paramet

- **ctx**
se-
cu-
rity
con-
text
- **dep**
the
de-
ploy
to
be
val-
i-
date
- **prop**
the
list
of
im-
age
meta
prop
to
be
val-
i-
date

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if:
*

tion for accessing image failed; * HEAD request to image URL failed or returned response code != 200;
* HEAD request response does not contain Content-Length header; * the protocol specified in image
URL is not supported.

ties.

ironic.drivers.modules.fake module

con-
nec-
tion
to
glan
faile
*
au-
tho-
riza-

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
the
im-
age
does
con-
tain
the
men
tion
prop
er-

Fake
driv
in-
ter-
face
used
in
test-
ing.
This
is
also

stance, the `MultipleVendorInterface` class demonstrates how to load more than one interface and wrap them in some logic to route incoming `vendor_passthru` requests appropriately. This can be useful eg. when mixing functionality between a power interface and a deploy interface, when both rely on separate `vendor_passthru` methods.

an
ex-
am-
ple
of
som
kind
of
thing
whic
can
be
done
with
driv
For
in-

class ir

Base
iro
dri
bas
BIO

Fake
im-
ple-
men
ta-
tion
of
sim-
ple
BIO
In-
ter-
face

apply_co

Val-
i-
date
&
ap-
ply

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

BIO
set-
tings
on
the
give
node

This
meth
take
the
BIO
set-
tings
from
the
set-
tings
para
and
ap-
plies
BIO
set-
tings
on
the

Paramete

- **task**
a
Task
ager
in-
stan
- **set**
Dic-
tona
con-
tain-
ing
the
BIO

tion.

con-
fig-
u-
ra-
tion.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
BIO
con-
fig-
u-
ra-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
set-
ting
fails

Raises

Miss
ing-
Pa-
ram-
e-
ter-

Valu
if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

plete.

cache_b:

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
store

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

ties from bare metal.

BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing
clea
ing
op-
er-
a-
tion

Paramete
task

a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
get-
ting
BIO
prop
er-

Returns

Non

factory_

Re-
set
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node
Af-

ter the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

Paramet
task

a
Task
ager
in-
stan

Raises
Un-

plete.

sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
BIO
re-
set.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This
meth
is

long-running checks.

of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss

ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

class ir

Base
iro
dri
bas
Boo

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sim-
ple
boot
in-
ter-
face

capabil:

clean_up

Clea
up
the
boot
of
in-
stan
This

meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

Parame

task

A
task
from
Task
ager

Returns

Non

clean_up

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot

cue ramdisk.

ing
the
de-
ploy
or
res-

Parame

task

A
task
from
Task
ager

Returns

Non

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare_

Pre-
pare
the
boot
of

tion from the nodes database.

in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parame

tas
A
task
from
Task
ager

Returns

Non

prepare.

Pre-
pare
the
boot
of
Iron
ram

This
meth
pre-
pare
the
boot
of
the

vant information from the nodes database.

might want to boot the ramdisk in different ways by passing parameters to them. For example,

de-
ploy
or
res-
cue
ramd
af-
ter
read
ing
rel-
e-

Parameter

- **task**
A
task
from
Task
ager
- **ramd**
The
op-
tion
to
be
pass
to
the
iron
ramd
Dif-
fer-
ent
im-
ple-
men
ta-
tion

Whe
Age
ramd
is
boot
to

etc.

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

de-
ploy
a
node
it
take
the
pa-
ram-
e-
ters
ipa-
api-
url,

Othe
im-
ple-
men
ta-
tions
can
mak
use
of
ram
to
pass
such
in-
for-
ma-
tion.
Dif-
fer-

Returns

Non

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men

the required information for this interface to function.

long-running checks.

info.
This
meth
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame
task
A
Task
ager
in-
stan

con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iron
dri
bas
Con

Ex-
am-
ple
im-

ple-
men-
ta-
tion
of
a
sim-
ple
con-
sole
in-
ter-
face

get_console

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for
the
clien
to
ac-
cess
the

console.

Parameters

tries

start_co

Star

a

re-

mote

con-

sole

for

the

task

node

This

meth

shou

not

raise

an

ex-

cep-

tion

if

con-

sole

al-

read

start

Parame

task

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

stop_co

Stop

the

re-

mote

con-

sole
ses-
sion
for
the
task
node

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

validat

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of

the required information for this interface to function.

long-running checks.

the
task
node
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
bas
Dep

Clas
for
a
fake
de-
ploy
men
driv

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
de-
ploy

rate power interface.

in-
ter-
face
that
uses
a
sep-
a-

clean_up

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

If
im-
ple-
men

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
tear

Parame

task

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

deploy(*t*)

Per-
form
a
de-
ploy
men
to
the

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto
the
spec-
i-
fied
node
This

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy

this method should be implemented by the driver.

the same node on the same conductor.

men
en-
vi-
ron-
men-
ahead
of
time
is
pos-
si-
ble,

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
*de-
ploy*

Paramet

task
A
Task
ager
in-
stan
con-

tain-
ing
the
node
to
act
on.

take_ove

Take
over
man
age-
men
of
this
task
node
from
a
deac
con-
duc-
tor.

If
con-
duc-
tors
host
main
tain
a
stati
re-
la-
tion-
ship
to
node
this
meth
shou
be
im-

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

For exam
Neu
tron
mus

tftpboot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

for-
ward
DHCP
BOOT
re-
ques
to
a
con-
duc-
tor
whic
has
pre-
pare
the

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

tear_down

Tear
down
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node

sary to un-deploy that node.

Give
a
node
that
has
been
pre-
vi-
ously
de-
ploy
to,
do
all
clea
and
tear
dow
nec-
es-

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

validate

Val-
i-

the required information for this interface to function.

date
the
drive
spec
Nod
de-
ploy
men
info.
This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

long-running checks.

Parameter

task

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Raises

In-

valid

Pa-

ram-

e-

ter-

Value

on

mal-

form

pa-

ram-

e-

ter(s)

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Value

on

miss

ing

pa-

ram-

e-

ter(s)

class ir

Base

iron
dri
bas
Ins
Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
a
sim-
ple
in-
spec
in-
ter-
face

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect_

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tiona
hard
ware
prop
er-
ties.

Parame
task
A
task
from
Task
ager

Raises
Hard
ware
spec
tion-
Fail-
ure,
if
un-
able
to
get
es-
sen-
tial
hard
ware
prop

er-
ties.

Returns

Re-
sult-
ing
state
of
the
in-
spec
tion
i.e.
state
or
Non

validate

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

long-running checks.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

class ir

Base
iro
dri
bas
Man

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sim-
ple
man
age-
men
in-
ter-
face

get_boot

Get
the
cur-
rent
boot
de-

vice
for
a
node
Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa
that
not
all
driv
sup-
port
this.

Parame

task
A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
iro
com
boo
or
Non
if
it
is
un-
know

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

unknown.

get_ind:

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard
ware
com-
po-
nent

Parame

- **task**
A
task
from
Task
ager
- **comp**
The
hard
ware
com-
po-
nent
one
of
[ironic](#)
[com](#)
[com](#)
- **ind**
In-
di-
ca-
tor
ID
(as
re-

is specified.

port
by
get_

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
or
in-
di-
ca-
tor

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Cur-
rent
state
of
the
in-

di-
ca-
tor,
one
of
iro
com
ind

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sens

Get
sen-
sors
data
meth

Paramet

task
A
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data
fails

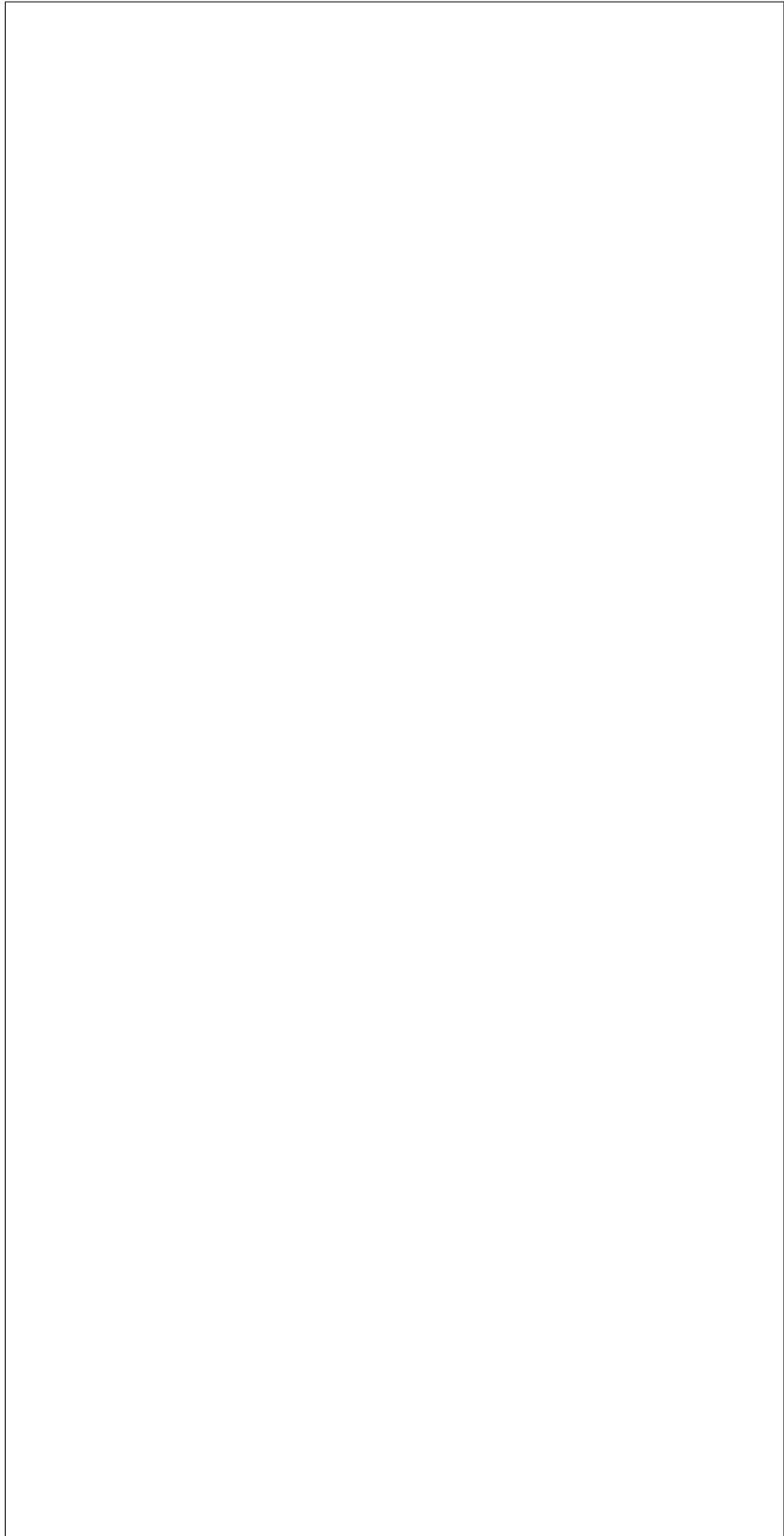
Raises

Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Returns

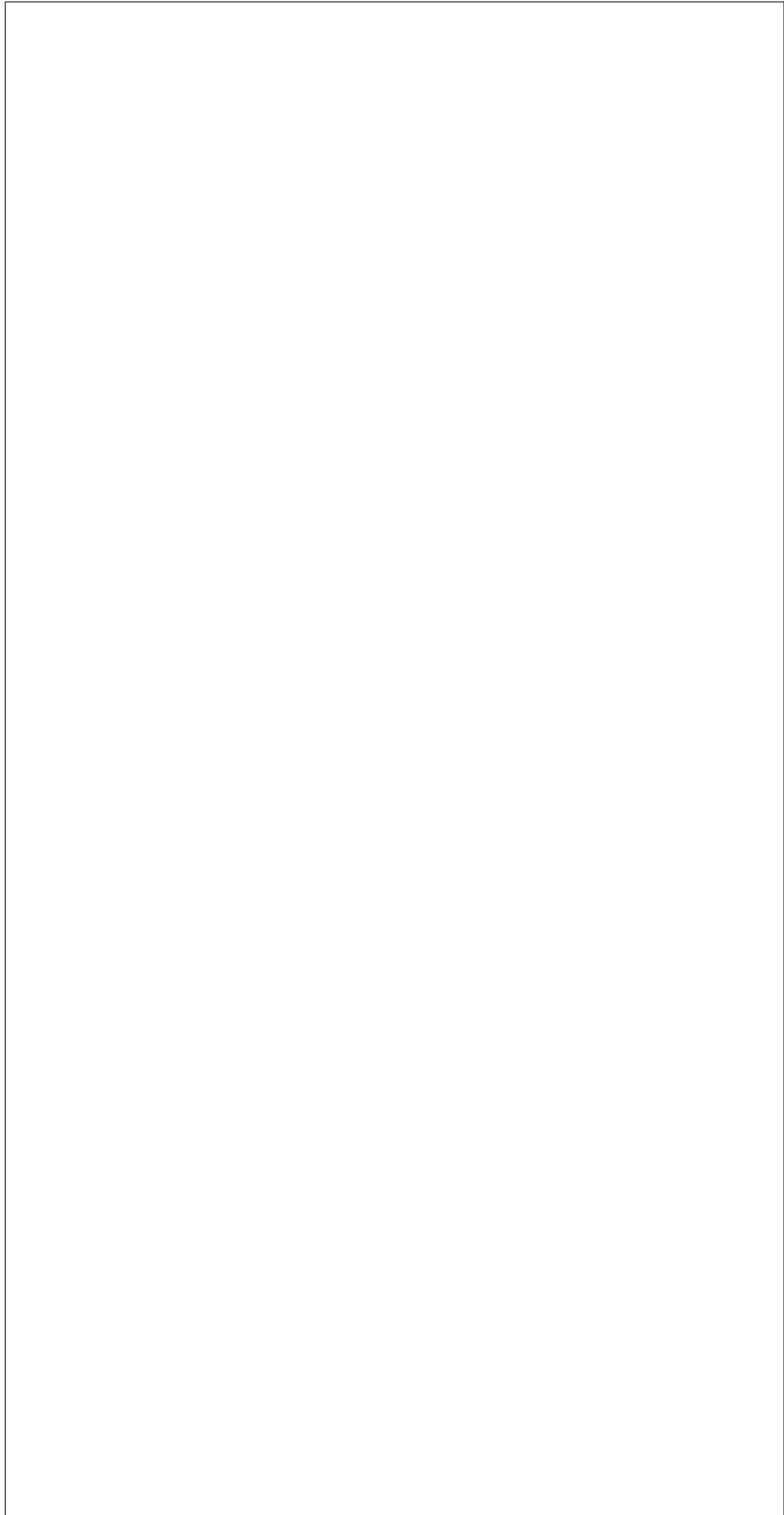
Re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic
can

be processed by Ceilometer. eg,



(continues on next page)

(continued from previous page)



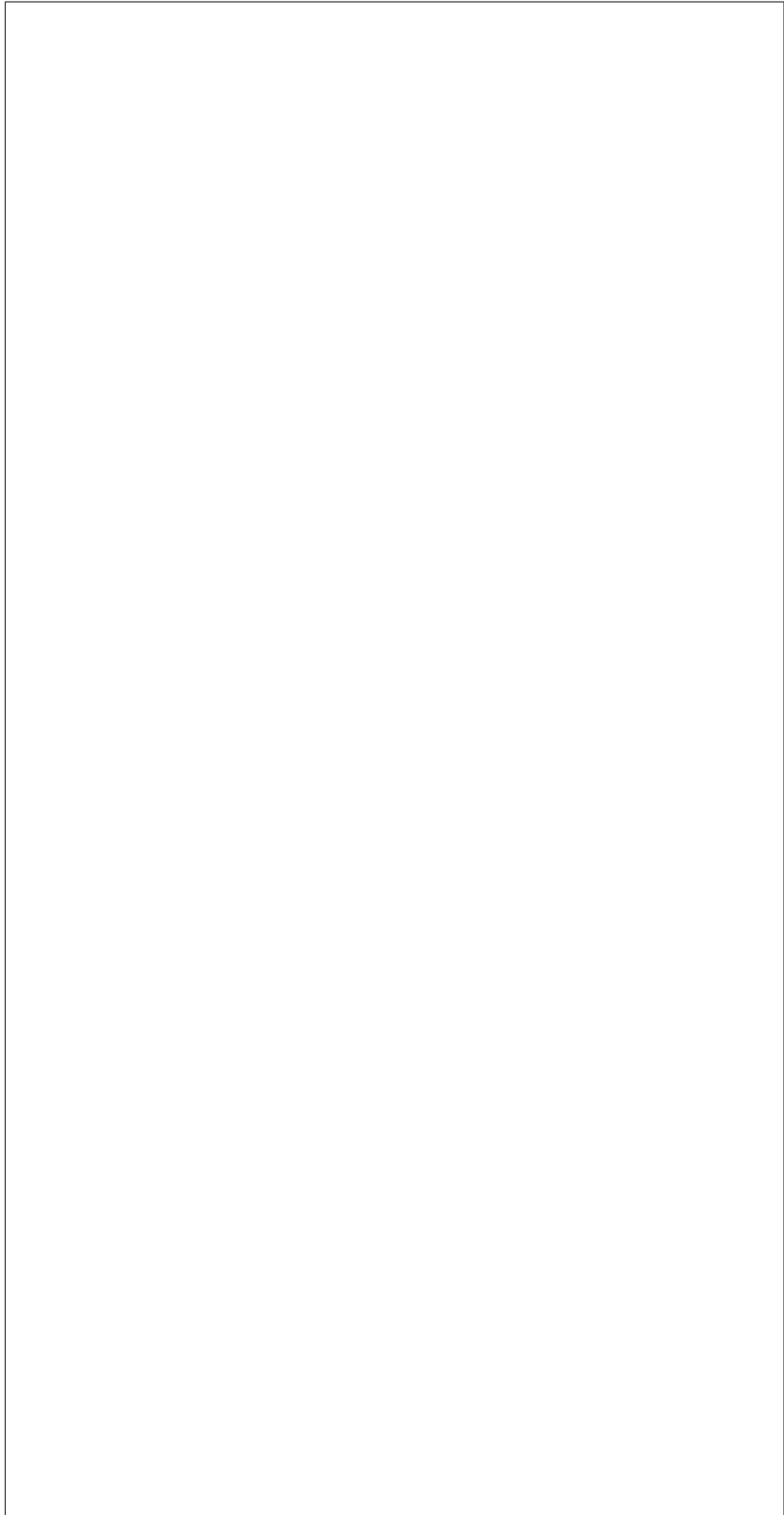
(continues on next page)

(continued from previous page)



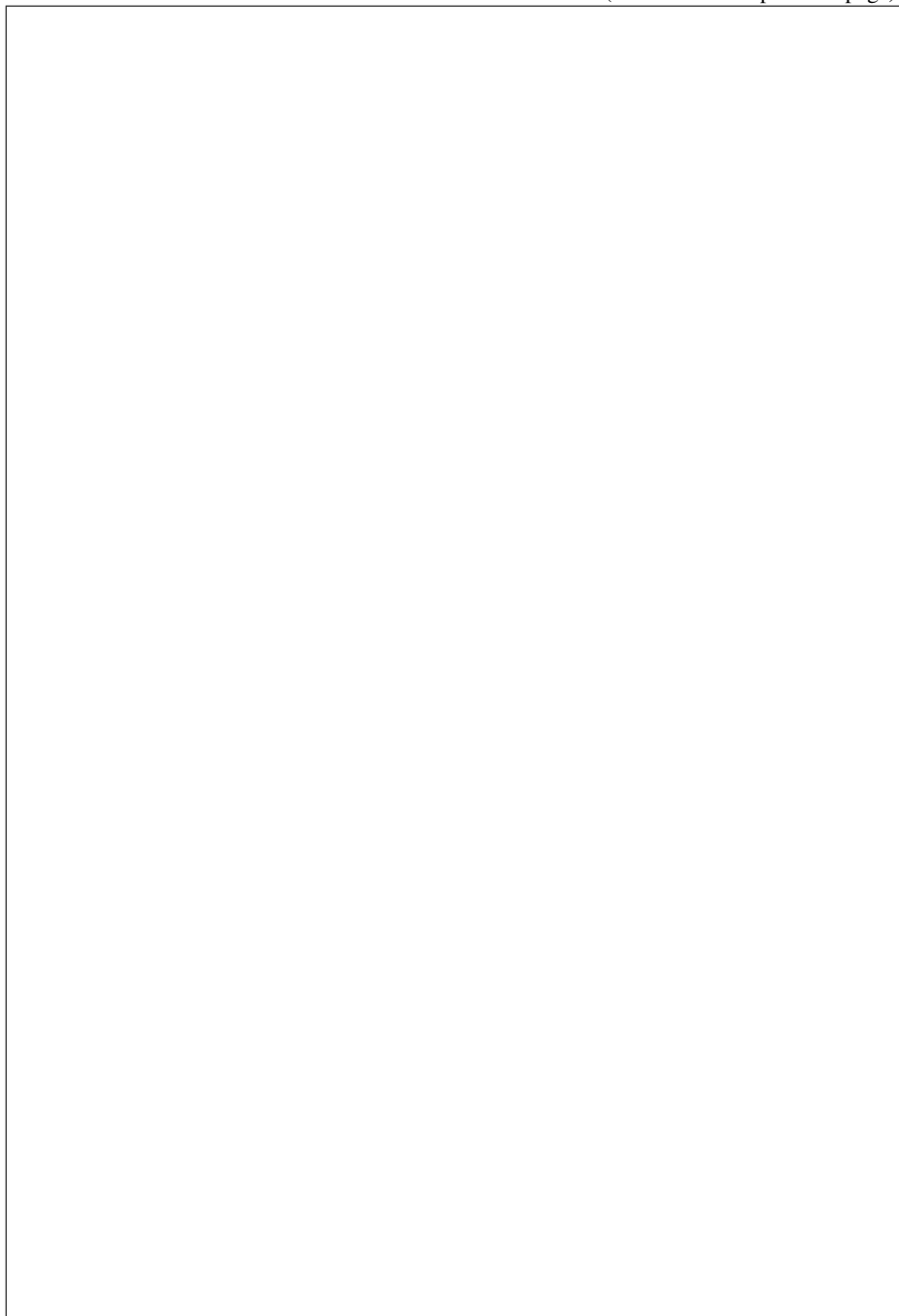
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter

task

A task from Taskager

Returns

A list with the supported boot device defined in *ironic-compute-boo*

get_supported

Get a map of the supported in-distribution (e.g. LED)

Parameter

- **task**

A task from Taskager

- **comp**

ment, otherwise return indicators for all existing components.

dictionaries having indicator IDs as keys and indicator properties as values.



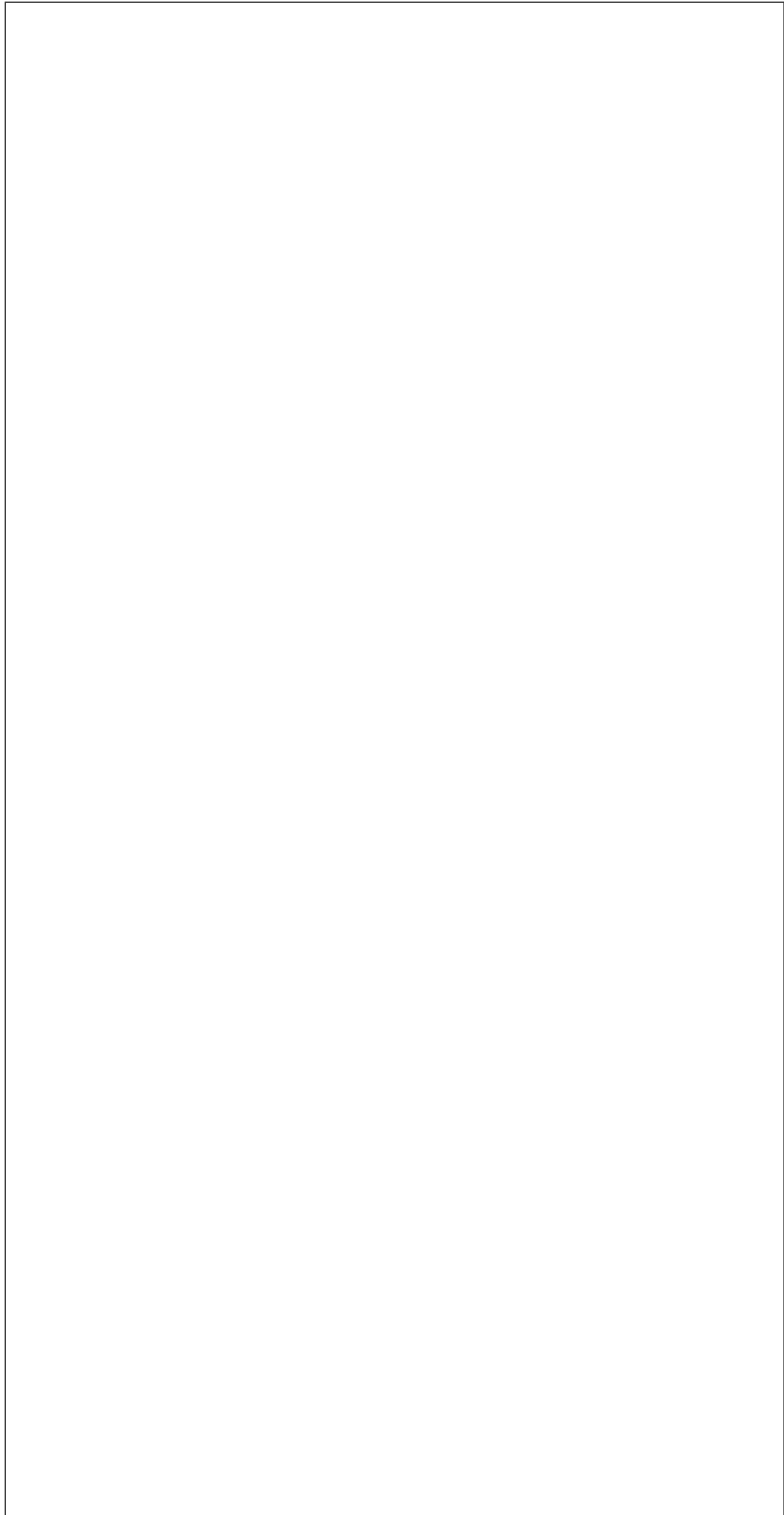
(continues on next page)

If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-
tion
for
just
this
com
po-

Returns

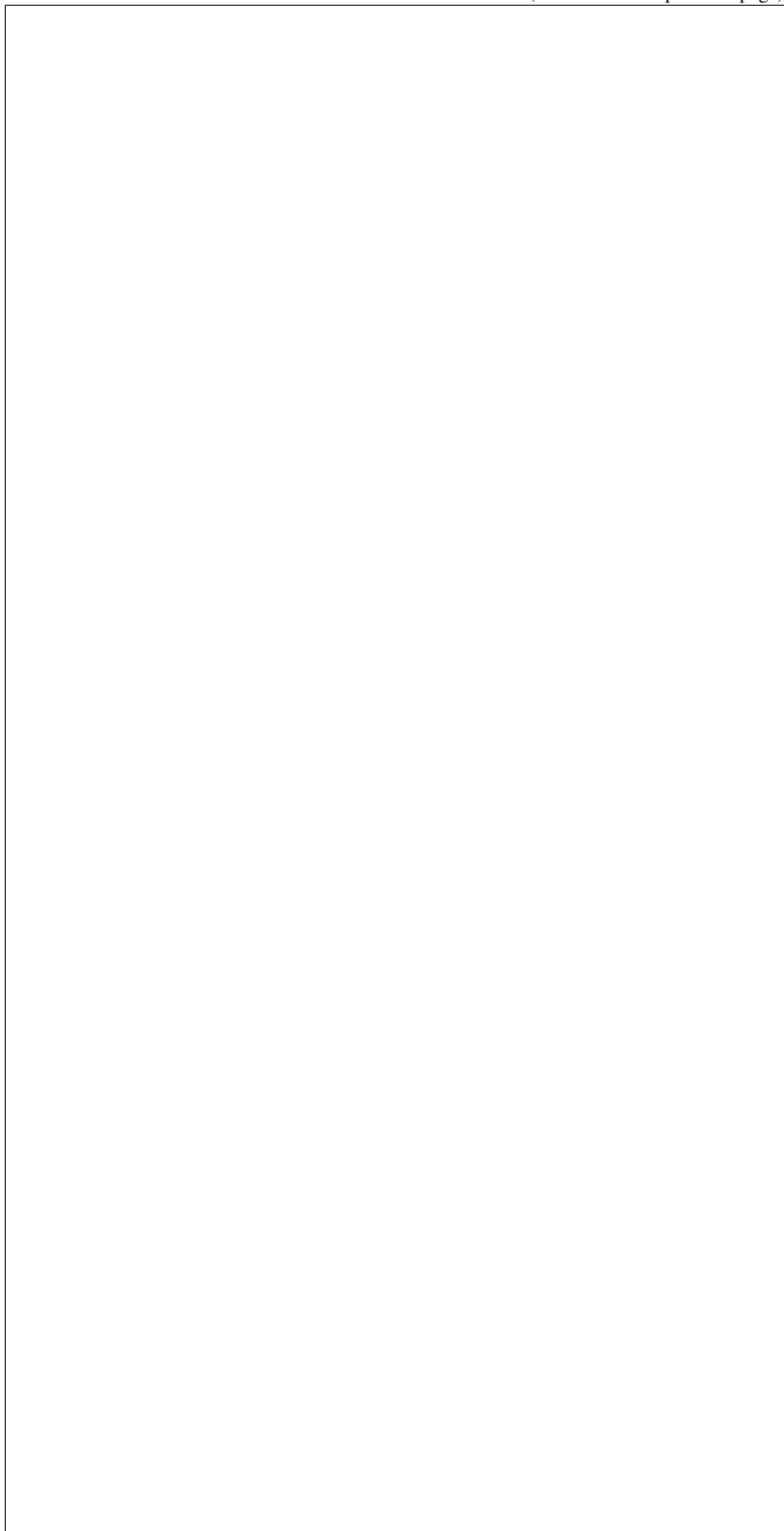
A
dic-
tio-
nary
of
hard
ware
com
po-
nent
(*irc*
com
com
as
keys
with
val-
ues
be-
ing

(continued from previous page)



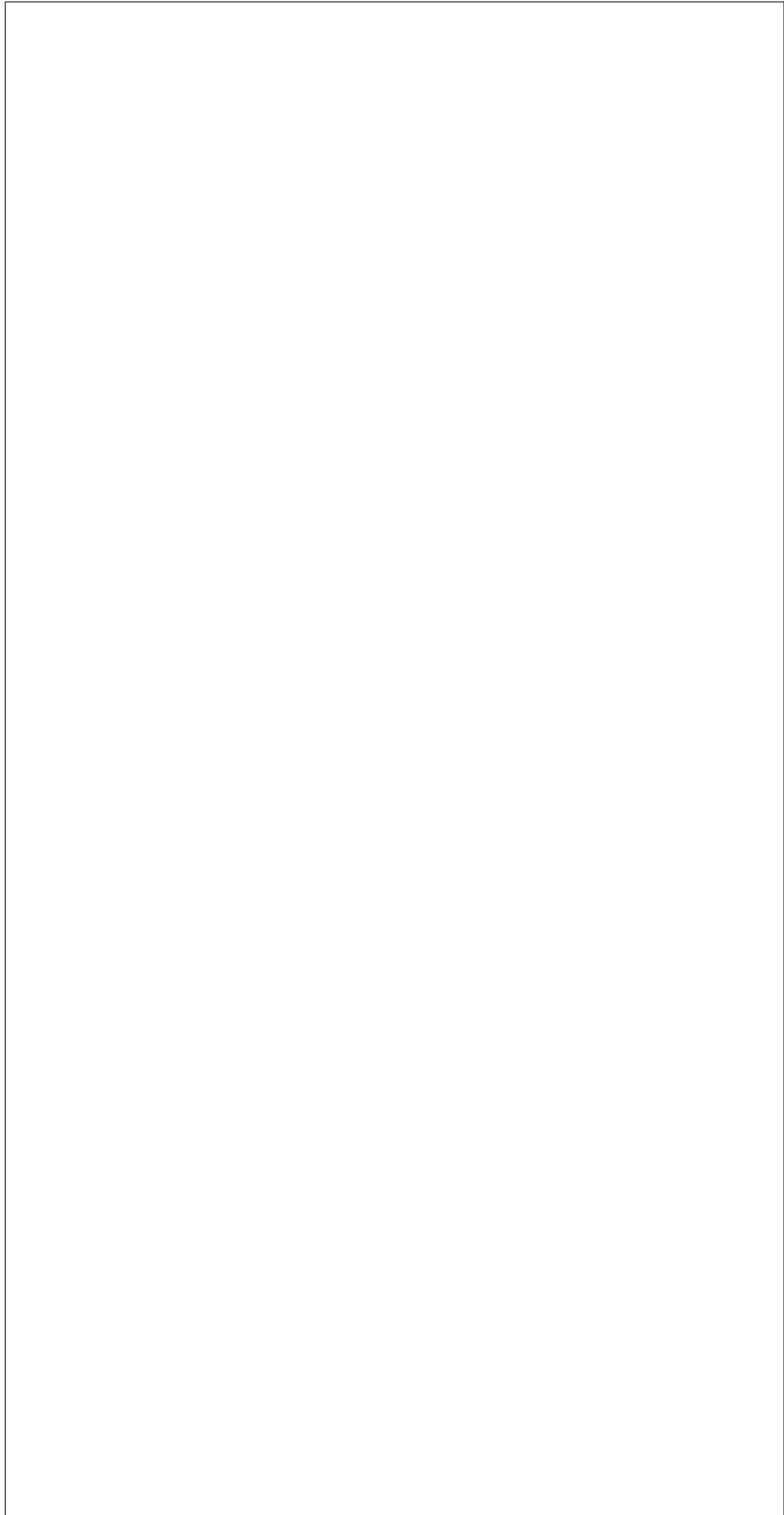
(continues on next page)

(continued from previous page)



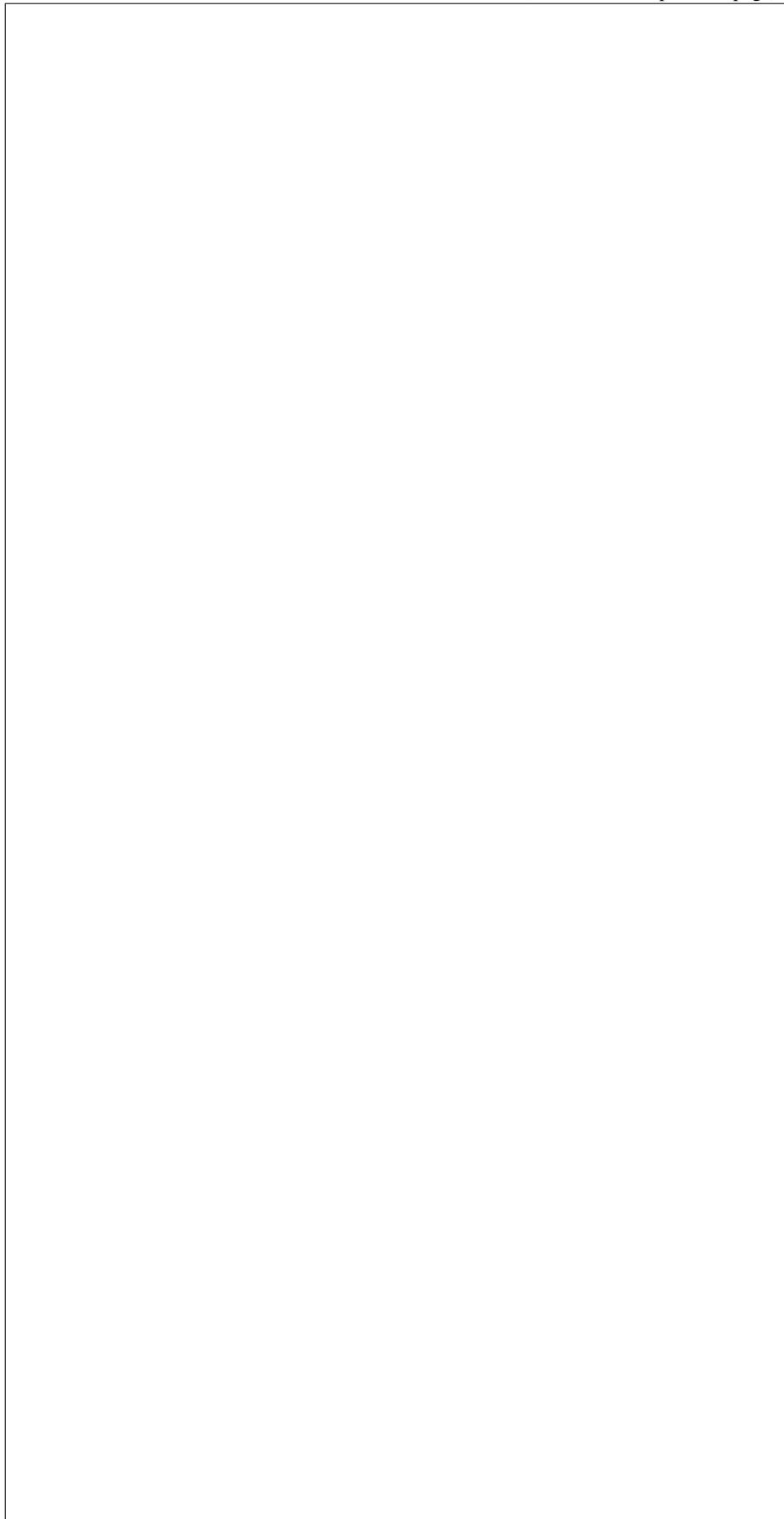
(continues on next page)

(continued from previous page)



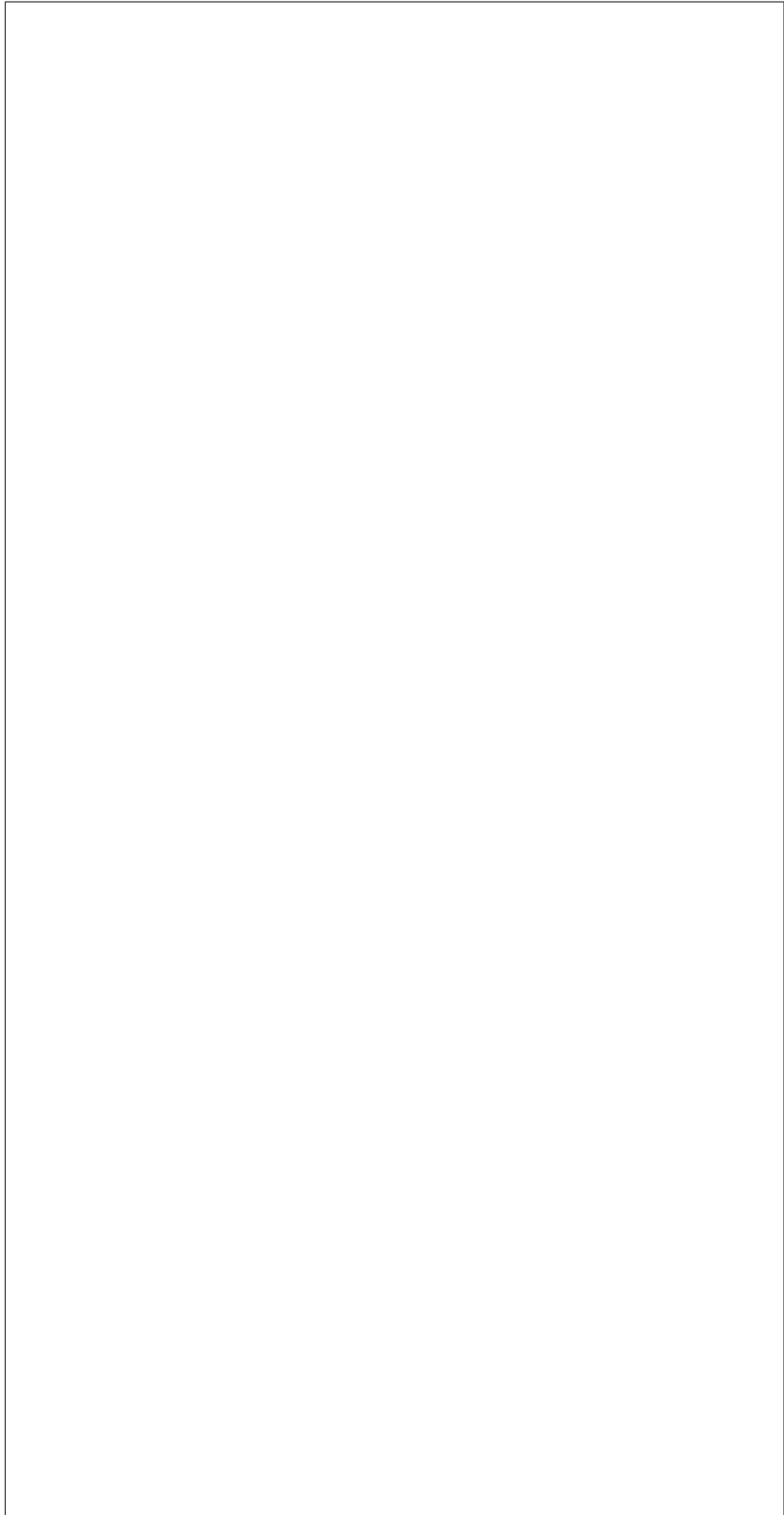
(continues on next page)

(continued from previous page)



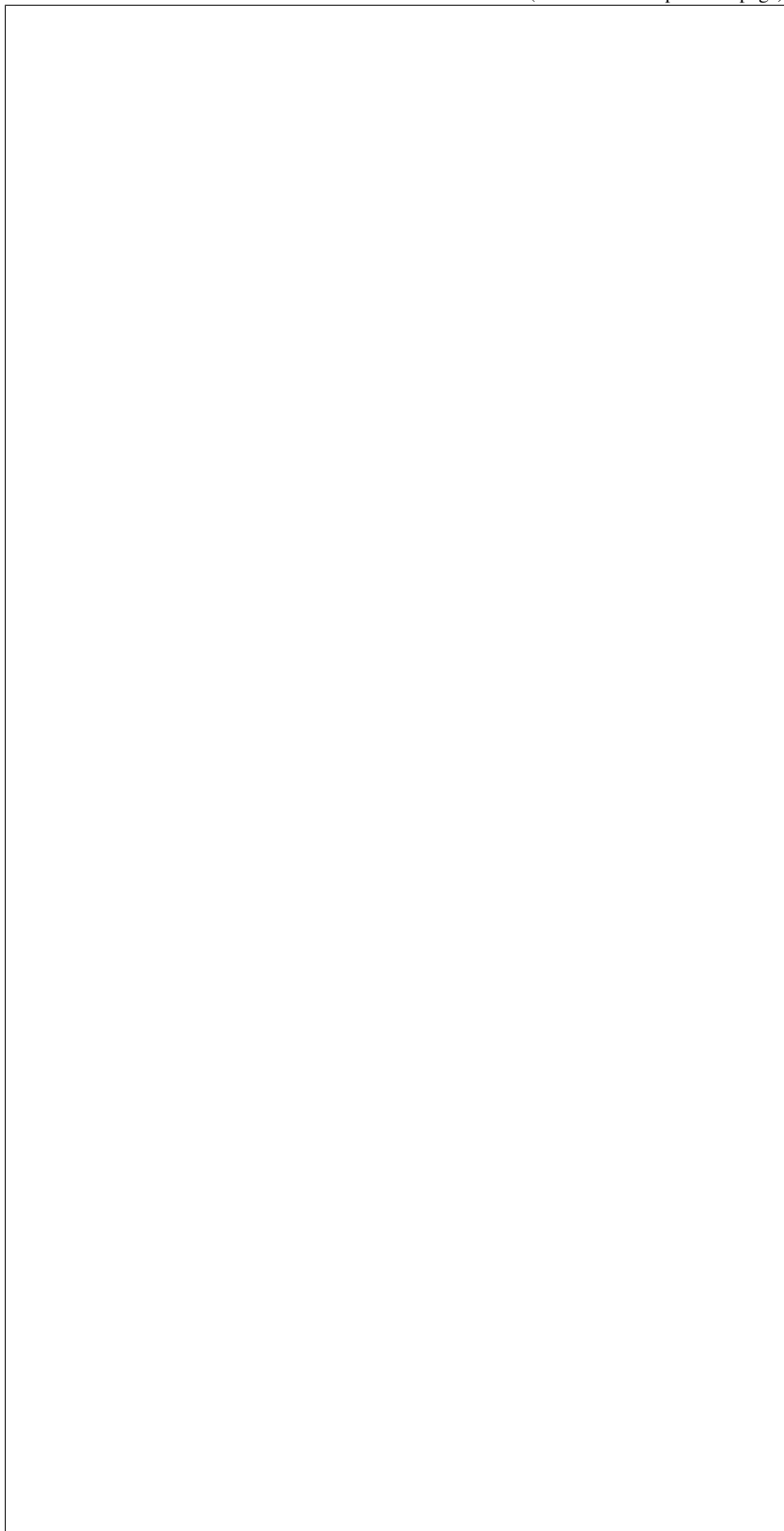
(continues on next page)

(continued from previous page)



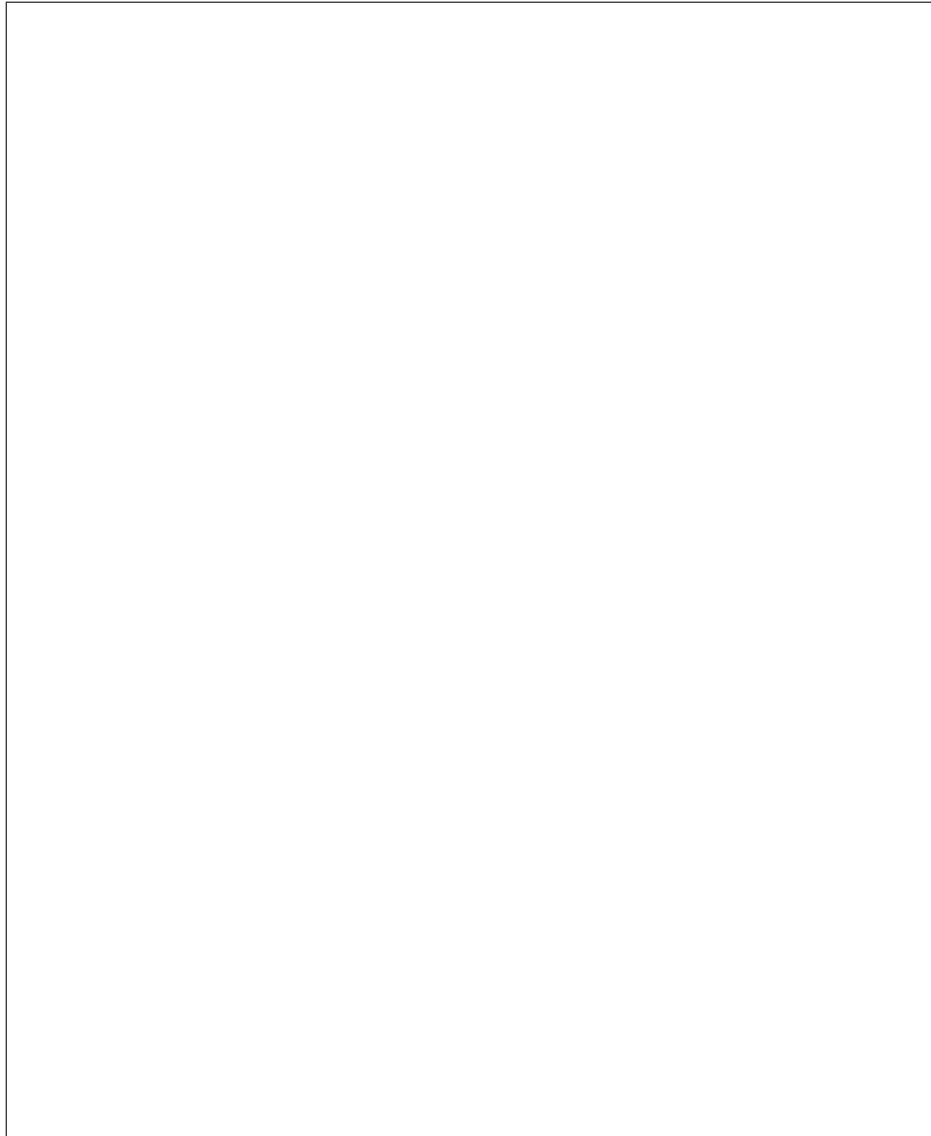
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



set_boot

Set the boot device for a node

Set the boot device to use on

next
re-
boot
of
the
node

Parameters

- **task**
A task from Taskager.
- **dev**
The boot device one of *ironic.com* *boot*
- **persist**
Boolean value True if the boot device will persist to all future boots. False if

not. Default: False.

Raises
In-

valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.
This

the required information for this interface to function.

long-running checks.

meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Paramet
task
A
Task
ager
in-
stan
con-
tain-

ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
bas
Pow
Ex-
am-
ple
im-
ple-
men

ta-
tion
of
a
sim-
ple
pow
in-
ter-
face

get_pow

Re-
turn
the
pow
state
of
the
task
node

Param

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-

ram-
e-
ter
is
miss
ing.

Returns

A
pow
state
One
of
iron
com
sta

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list
of
the
sup-
port

pow
state

Parameters

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
pow-
state
de-
fined
in
ironic
command
state

reboot (task)

Per-
form
a
hard
re-
boot
of
the
task
node.

Drivers
are
ex-
pect
to

it on.

prop
erly
han-
dle
case
whe
node
is
pow
ered
off
by
pow
er-
ing

Parame

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any

indicates to use default timeout.

pow
state
Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

set_pow

Set
the
pow
state
of
the
task
node

Paramet

- **task**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

- **power**
Any
power
state
from
ironic
command
state

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
power
state
Non

indicates to use default timeout.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

the required information for this interface to function.

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
nod
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it

long-running checks.

shou
not
con-
duct

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iron
dri
bas
RAI

Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
sim-
ple
RAI
In-
ter-
face

create_c

Cre-
ates
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

ified in the nodes `target_raid_config`. Default value is `True`.

give
node
It
as-
sum
that
the

Parameters

- task**
A
Task
ager
in-
stan
- create_root_volume**
Set-
ting
this
to
False
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- create_root_volume**
Set-
ting
this
to
False

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

creating the new configuration.

in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-
ra-

chronously, or None if it is complete.

ration is deleted, `node.raid_config` should be cleared by the implementation.

tion
is
in
prog
asyn

delete_

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
Af-
ter
RAI
con-
fig-
u-

Parame

task
A
Task
ager
in-
stan

Returns

state

it is complete.

(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro
or
Non
if

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

class ir

Bas
iro
dri

bas
Res
Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
a
sim-
ple
res-
cue
in-
ter-
face

get_prop

Re-
turn
the
prop-
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro-
erty
nam-
de-
scrip-
tion:
en-
tries

rescue()

Boo
the

task
node
into
a
res-
cue
en-
vi-
ron-
men

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
stan-
cue-
Fail-
ure
if
node
val-
i-
da-
tion
or
res-
cue
op-
er-
a-
tion
fails

Returns

state
if
res-
cue

is
in
prog
asyn
chro
or
state
if
it
is
com
plete

unrescue

Tear
down
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to
nor-
mal.

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
stan
Un-
res-
cue-

ation fails.

Fail-
ure
if
node
val-
i-
da-
tion
or
un-
res-
cue
op-
er-

Returns

state
if
it
is
suc-
cess
ful.

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.
This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties

the required information for this interface to function.

long-running checks.

of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-

Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
bas
Sto

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
sim-
ple
stor-
age
In-
ter-
face

attach_v

In-

form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

task
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

detach_v

In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parameter

task

A

Task

ager

in-

stan

Raises

Un-

sup-

port

ed-

Drive

ten-

sion

get_prop

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

should_v

De-

ter-

min

if

de-

ploy

shou

per-
form
the
im-
age
writ
out.

Parame

task

A
Task
ager
in-
stan

Returns

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect
the
im-
age
to
be
writ

ten by Ironic.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
the

the required information for this interface to function.

long-running checks.

drive
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
bas

Ven
Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
ven-
dor
pass
in-
ter-
face

first_me

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat

Val-
i-

date
vend
spec
ac-
tion:
If
in-
valid
raise
an
ex-
cep-
tion:
oth-
er-
wise
re-
turn
Non

Parame

- **task**
A
task
from
Task
ager
- **metl**
Metl
to
be
val-
i-
date
- **kwa**
Info
for
ac-
tion.

Raises
Un-
sup-
port
ed-
Driv

faces.

ten-
sion
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain
meth

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

class ir

Base
iro
dri
bas
Ven

Ex-
am-
ple
im-

ple-
men
ta-
tion
of
a
sec-
onda
ven-
dor
pass

fourth_r

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

second_r

third_me

validate

Val-
i-
date
vend

spec
ac-
tion

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **task**
A
task
from
Task
ager
- **met**
Met
to
be
val-
i-
date
- **kw**
Info
for
ac-
tion.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

faces.

ironic.drivers.modules.image_cache module

if
meth
can
not
be
map
to
the
sup-
port
in-
ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain
meth

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

Util-
ity
for
cach
mas
ter
im-
ages

class ir

Base
obj
Clas
han-
dling
ac-
cess
to
cach
for
mas
ter
im-
ages

clean_up

Clea
up
di-
rec-
tory
with
im-
ages
keep
ing
cach
of
the
lat-
est
im-
ages

Files
with
link
count
>1
are
never
dele
Pro-
tecte
by
glob
lock
so
that

ter images after we get listing and before we actually delete files.

reached, even if it is possible to clean up more files

no
one
mes
with
mas

Parame

am
if
pres
amo
of
spac
to
re-
clain
in
byte
clea
ing
will
stop
if
this
goal
was

fetch_in

Fetc
im-
age
by
give
href
to
the
des-
ti-
na-
tion
path

Doe
noth
ing
if
des-
ti-
na-
tion

contents. Only creates a hard link (`dest_path`) to cached image if requested image is already in cache and up to date with `href` contents. Otherwise downloads an image, stores it in cache and creates a hard link (`dest_path`) to it.

path
ex-
ists
and
is
up
to
date
with
cach
and
href

Parame

- **href**
im-
age
URI
or
href
to
fetch
- **dest**
des-
ti-
na-
tion
file
path
- **ctx**
con-
text
- **for**
bool
valu
whe
to
con-
vert
the
im-
age

to
raw
for-
mat

ironic.d

Ex-
plic-
itly
clea
cach
base
on
their
pri-
or-
ity
(if
re-
quir

This
clea
up
the
cach
to
free
up
the
amo
of
spac
re-
quir
for
the
im-
ages
in
im-

ages_info. The caches are cleaned up one after the other in the order of their priority. If we still cannot free up enough space after trying all the caches, this method throws exception.

Paramet

- **ctx**
con-
text
-

in cache.

dir
the
di-
rec-
tory
(of
the
cach
to
be
free
up.

- **ima**
a
list
of
tu-
ples
of
the
form
(im-
age_
for
whic
spac
is
to
be
cre-
ated

Raises

In-
suf-
fi-
cien
Disk
ex-
cep-
tion.
if
we
can-
not
free
up
enou
spac

ing all the caches.

`ironic.drivers.modules.image_utils` module

af-
ter
try-

`ironic.d`

Dec
o-
ra-
tor
meth
for
addi
clea
pri-
or-
ity
to
a
class

`class ir`

Base
obj

`publish`

Mak
im-
age
file
dow
load
able

De-
pend
ing
on
iron
set-
ting
push
give
file
into
Swi
or

servers document root and returns publicly accessible URL leading to the given file.

copied
it
over
to
local
cal
HTT

Parameters

- **image_path**
path to file to publish
- **object_name**
name of the published file

Returns

a URI to download published file

unpublished

Withdrawing the image previously made downloadable

published - Swift or local HTTP servers document root.

De-
pend
ing
on
iron
set-
tings
re-
mov
pre-
vi-
ousl
pub-
lishe
file
from
whe
it
has
been

Paramete

obj
nam
of
the
pub-
lishe
file
(op-
tiona

classmeth

With
draw
the
im-
age
pre-
vi-
ousl
mad
dow
load
able

De-
pend
ing
on
iron

published - Swift or local HTTP servers document root.

set-
tings
re-
mov
pre-
vi-
ousl
pub-
lishe
file
from
whe
it
has
been

Parame

- **node**
the
node
for
whic
im-
age
was
pub-
lishe
- **pre**
ob-
ject
nam
pre-
fix.
- **suf**
ob-
ject
nam
suf-
fix.

update_

ironic.d

Dele
the

im-
age
if
it
was
cre-
ated
for
the
node

Paramet

- **task**
an
iron
node
ob-
ject.
- **pre**
Pre-
fix
to
use
for
the
ob-
ject
nam

ironic.d

Dele
the
flopp
im-
age
if
it
was
cre-
ated
for
the
node

Paramet

task
an
iron

node
ob-
ject.

ironic.d

Dele
the
ISO
if
it
was
cre-
ated
for
the
in-
stan

Paramet

task
A
task
from
Task
ager

ironic.d

ironic.d

Pre-
pare
boot
ISO
im-
age

Buil
boot
ISO
out
of
[in-
stan
[in-
stan
and
[dri
if
pres
Oth-
er-
wise

ramdisk_id from *[instance_info]/image_source* Glance image metadata.

age.

read
ker-
nel_
and

Push
pro-
duce
ISO
im-
age
up
to
Glan
and
re-
turn
tem-
po-
rary
Swi
URI
to
the
im-

Paramet

- **task_id**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **d_i**
De-
ploy
men
in-

for-
ma-
tion
of
the
node

- **root**
Root
URI

Returns

boot
ISO
HTT
URI

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss

ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-

ram-
e-
ters
have
in-
valid
valu

Raises

Im-
age-
Cre-
ation
Failure
if
cre-
at-
ing
ISO
im-
age
faile

ironic.d

Pre-
pare
an
im-
age
with
con-
fig-
driv

De-
code
base
con-
tents
and
write
it
into
a
disk
im-
age
that
can
be
at-
tach

a virtual USB device. Images stored in Swift are downloaded first.

e.g.
to

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **con**
Con
fig
driv
as
a
base
enco
strin

Raises
Im-
age-
Cre-
ation
Fail-
if
it
faile
whil
cre-
at-
ing
the
im-
age.

Raises
Swi
Op-

er-
a-
tion)
if
any
op-
er-
a-
tion
with
Swi
fails

Returns

im-
age
URI
for
the
im-
age.

ironic.d

Pre-
pare
de-
ploy
or
res-
cue
ISO
im-
age

Buil
boot
ISO
out
of
[dri
or
[dri
and
[dri
then
push
built
im-
age
up
to
Glar

turn temporary Swift URL to the image.

will be written into an appropriate location on the final ISO.

and
re-

If
net-
worl
in-
ter-
face
sup-
plies
net-
worl
con-
fig-
u-
ra-
tion
(*net-
worl*
a
*net-
worl*

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **par**
a
dic-
tio-
nary
con-
tain-

to kernel command line.

ing
pa-
ram-
e-
ter
nam
>val
map
ping
to
be
pass

- **mod**
ei-
ther
de-
ploy
or
res-
cue.

- **d_i**
De-
ploy
men
in-
for-
ma-
tion
of
the
node

Returns
boot
ISO
HTT
URI

Raises
Mis
ing-
Pa-
ram-
e-
ter-
Valu
if
any

ing.

of
the
re-
quir
pa-
ram-
e-
ters
are
miss

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

Raises

Im-
age-
Cre-
ation
Fail
if
cre-
at-
ing
ISO
im-
age
faile

ironic.d

Pre-
pare
an

im-
age
with
the
give
con-
tent.

If
con-
tent
is
al-
read
an
HTT
URI
re-
turn
it
un-
char

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **con**
Con
tent
as
a
strin
with
a
file
nam

or
byte
with
con-
tents

- **pre-**
Pre-
fix
to
use
for
the
ob-
ject
nam

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the
im-
age.

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi
fails

Returns

im-

age
URI
for
the
im-
age.

ironic.d

Pre-
pare
the
flopp
im-
age
for
pass
ing
the
pa-
ram-
e-
ters.

This
meth
pre-
pare
a
tem-
po-
rary
VFA
files
tem
im-
age
and
adds
a
file
into
the
im-

age which contains parameters to be passed to the ramdisk. Then this method uploads built image to Swift [redfish]swift_container, setting it to auto expire after [redfish]swift_object_expiry_timeout seconds. Finally, a temporary Swift URL is returned addressing Swift object just created.

Paramet

- task

to deploy or rescue image via floppy image.

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **par:**
a
dic-
tio-
nary
con-
tain-
ing
pa-
ram-
e-
ter
nam
>val
map
ping
to
be
pass

Raises
Im-
age-
Cre-
ation
Fail-
if
it
faile
whil
cre-
at-
ing
the
flopp
im-

age.

Raises

Swi
Op-
er-
a-
tionl
if
any
op-
er-
a-
tion
with
Swi
fails

Returns

im-
age
URI
for
the
flopp
im-
age.

`ironic.drivers.modules.inspect_utils` module

`ironic.d`

Cre-
ate
iron
port
from
MA
ad-
dres
data
dict.

Cre-
ates
iron
port
from
MA
ad-
dres

operator. Helper argument to detect the MAC address `get_mac_address` defaults to value part of MAC address dict key-value pair.

Parameter

- **task**
A Task manager instance
- **mac**
A dictionary of MAC addresses returned by node inspection.
- **get**
a function to get the MAC

key-value pair of the previous `macs` argument.

`ironic.drivers.modules.inspector` module

ad-
dres
from
mac
item
A
mac
item
is
the
dict

Modules r

<https://pypi.org/project/ironic>

class ir

Base
ironic
driver
base
Inspector

In-
band
in-
spec
tion
via
ironic
insp
proj

abort(*ta*

Abor
hard
ware
in-
spec
tion.

Parame

task

a
task
from
Task
ager

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware
to
ob-
tain
the
hard
ware
prop
er-
ties.

This
par-
tic-
u-

sults will be checked in a periodic task.

lar
im-
ple-
men-
ta-
tion
only
start
in-
spec
tion
us-
ing
iron
insp
Re-

Parame
task
a
task
from
Task
ager

Returns
state

Raises
Har
ware
spec
tion-
Fail-
ure
on
fail-
ure

validat

Val-
i-
date
the
driv
spec
in-
spec
tion
in-
for-

ma-
tion.
If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame
task
a
task
from
Task
ager

Raises
Un-
sup-
port
ed-
Driv
ten-
sion

ironic.drivers.modules.ipmitool module

IPM
pow
man
ager
driv
Uses
the
ip-
mi-
tool
com
man
(http
//

includes setting the boot device, getting a serial-over-LAN console, and controlling the power state of the machine.

PROVIDES DIFFERENT COMMAND-LINE OPTIONS AND *IS NOT SUPPORTED* BY THIS DRIVER.

ipmi
sour
net/
to
re-
mote
man
age
hard
ware
This

NOT
THA
CER
TAL
DIS
TRC
MA
IN-
STA
oper
BY
DE-
FAU
IN-
STE
OF
ip-
mi-
tool.
WH

class ir

Base
iro
dri
bas
Con

A
base
Con
sole
ter-
face
that
uses

ip-
mi-
tool.

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validate

Val-
i-
date
the
Nod
con-
sole
info.

Paramete

task
a
task
from
Task
ager

Raises

In-

valid
Pa-
ram-
e-
ter-
Valu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
whe
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

class ir

Base
iro
dri
bas
Man

detect_v

De-
tect
and
re-
turn
the
hard
ware
ven-
dor.

Paramet

task
A
task
from

state is specified.

Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
in-
di-
ca-
tor
or

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Strin
rep-
re-
sent
ing
the
BM

returns None.

re-
port
Ven-
dor
or
Man
u-
fac-
ture
oth-
er-
wise

get_boot

Get
the
cur-
rent
boot
de-
vice
for
the
task
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Param

task
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram

e-
ter-
Valu
if
re-
quir
IPM
pa-
ram-
e-
ters
are
miss
ing.

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

a

dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
iron
comm
boo
or
Non
if
it
is
un-
know

persist

When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

unknown.

get_prop

Re-
turn
the
prop

er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sens

Get
sen-
sors
data

Parame

task
a
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data
fails

Raises

Fail
ToP
eSer
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

re-
turn
a
dict
of
sen-
sor
data
group
by
sen-
sor
type

get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

task
a
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
[ironic](#)
[comm](#)
[boo](#)

inject_r

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

IP-
MI-
Fail-
ure
on
an
er-

ror
from
ip-
mi-
tool.

Returns

Non

set_boot

Set
the
boot
de-
vice
for
the
task
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Paramet

- **task**
a
task
from
Task
ager
- **dev:**
the
boot
de-
vice
one

not. Default: False.

of
iro
com
boo

- **per:**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied

Raises

Mis-
ing-
Pa-
ram-
e-

ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises
IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

validate

Che
that
driv
con-
tain
IPM
cre-
den-
tials
Val-
i-
date
whe
the
driv
prop
erty
of
the
sup-
plie
task

dentials information.

node
con-
tains
the
re-
quir
cre-

Parame
tas
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
IPM
pa-
ram-
e-
ters
are
miss
ing.

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter

is
miss
ing.

class *ir*

Base
iro
dri
bas
Pow

get_*pow*

Get
the
cur-
rent
pow
state
of
the
task
node

Param

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

one
of
iron
POV
POV
or
ER-
ROF

Raises

In-

valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool
(from

_pow
call)

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_supp

Get
a
list
of
the
sup-
port
pow
state

Paramet

task
A
Task
ager
in-
stan-
con-
tain-
ing
the

node
to
act
on.
cur-
rent
not
used

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
iro
com
sta

reboot()

Cy-
cles
the
pow
to
the
task
node

Paramete

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

timeout is counted once during power off and once during power on for reboots. None indicates that the default timeout will be used.

on.
•
time
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
The

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises

Power-
State
Failure
if
the
final
state
of
the
node
is
not
POWER
or
the
in-

intermediate state of the node is not POWER_OFF.

set_power

Turn
the
pow
on,
off,
soft
re-
boot
or
soft
pow
off.

Parameters

- task

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **power**
de-
sired
power
state
one
of
iron:
POW
POW
SOF
or
SOF

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
power
state
The

timeout is counted once during power off and once during power on for reboots. None indicates that the default timeout will be used.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Pow
er-
State
Fail-
ure
if
the
pow
coul
be
set
to
psta

validate

Val-
i-
date
drive
for
ip-
mi-
tool
drive

Che
that
node
con-
tain
IPM
cre-
den-
tials

Param

task
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-

ters
are
miss
ing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

class ir

Base
iro
dri
mod
ipm
IPM

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
shel
linal

get_cons

Get
the
type

and
con-
nec-
tion
in-
for
ma-
tion
about
the
con-
sole

start_co

Star
a
re-
mote
con-
sole
for
the
node

Parame

task
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

word

Raises
Pass
wor
File-
Fail
ToC
ate
if
un-
able
to
cre-
ate
a
file
con-
tain-
ing
the
pass

Raises
Con
sole
ror
if
the
di-
rec-
tory
for
the
PID
file
can-
not
be
cre-
ated

Raises
Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing

the
sub-
pro-
cess
faile

stop_con

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

task
a
task
from
Task
ager

Raises

Con
sole
ror
if
un-
able
to
stop
the
con-
sole

class ir

Base
iro
dri
mod
ipm
IPM

A
Con
sole
ter-

face
that
uses
ip-
mi-
tool
and
so-
cat.

get_cons

Get
the
type
and
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

Paramete

task
a
task
from
Task
ager

start_co

Star
a
re-
mote
con-
sole
for
the
node

Paramete

task
a
task
from

Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Pass
wor
File-
Faile
ToC
ate
if
un-
able
to
cre-
ate
a
file
con-
tain-
ing
the
pass

word

Raises

Con
sole
ror
if
the
di-
rec-

tory
for
the
PID
file
can-
not
be
cre-
ated

Raises

Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing
the
sub-
pro-
cess
faile

stop_con

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

task
a
task
from
Task
ager

Raises

Con
sole

ror
if
un-
able
to
stop
the
con-
sole

class ir

Base
iro
dri
bas
Ven

bmc_res

Re-
set
BMC
with
IPM
com
man
bmc
re-
set
(war

Parame

- **task**
a
Task
ager
in-
stan
- **http**
the
HTT
meth
used
on
the
re-
ques

-

warn
bool
pa-
ram-
e-
ter
to
de-
cide
on
warn
or
cold
re-
set.

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

In-

valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is
spec
i-
fied

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

send_raw

Sen
raw
byte
to
the
BM
Byte

shou
be
a
strin
of
byte

Parame

- **task**
a
Task
ager
in-
stan
- **http**
the
HTT
meth
used
on
the
re-
ques
- **raw**
a
strin
of
raw
byte
to
send
e.g.
0x00
0x0

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-

tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is
spec
i-
fied.

validate

Val-
i-
date
vend
spec
ac-
tions
If
in-
valid

raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Valid me

- send
- bmc

Parame

- **task**
a
task
from
Task
ager
- **metl**
meth
to
be
val-
i-
date
- **kwa**
info
for
ac-
tion.

Raises

In-
valid
Pa-
ram-
e-
ter-

fied.

Valu
whe
an
in-
valic
pa-
ram-
e-
ter
valu
is
spec
i-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.d

Che
if
the
com
man
stde
con-
tains
ci-
pher
suite
er-
rors.

Paramet

cmd.

The
com
man
stde

Returns

True
if
the
cmd
con-
tains
a
ci-
pher
suite
er-
ror,
False
oth-
er-
wise

`ironic.d`

Give
the
pos-
si-
ble
next
avai
ci-
pher
suite
ver-
sion

Base
on
CON
and
the
last
ci-
pher
suite
ver-
sion
used
that
faile

if the node doesn't have `cipher_suite` set. Starts using the last element of the list and decreasing the index.

This
func
tion
is
only
called

Parameter

act
lat-
est
ci-
pher
suite
used
in
the
ipmi
call.

Returns

the
next
pos-
si-
ble
ci-
pher
suite
or
None
in
case
of
emp
con-
fig-
u-
ra-
tion.

ironic.d

Dun
SDR
data
to
a
file.

Parameter

- **task**
a
Task
ager
in-
stan

- **file**
the
path
to
SDR
dum
file.

Raises
IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises
Mis-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises
In-

valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is
spec
i-
fied.

ironic.d

Send
raw
byte
to
the
BM
Byte
shou
be
a
strin
of
byte

Paramet

- **task**
a
Task
ager
in-
stan
- **raw**
a
strin
of
raw
byte
to
send
e.g.

0x00

0x01

Returns

a
tu-
ple
with
std-
out
and
stde

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

In-
valid
Pa-
ram-
e-

ter-
Valu
whe
an
in-
valid
valu
is
spec
i-
fied.

ironic.d

Up-
date
vari-
able
and
the
ci-
pher
suite
cmd

This
func
tion
up-
date
the
val-
ues
for
all
pa-
ram-
e-
ters
so
they
can
be
used
in

the next retry of `_exec_ipmitool`.

Paramet

- **act**
a

be modified in-place.

string
that
rep-
re-
sent
the
ci-
pher
suite
that
was
used
in
the
com
man

- **arg**
a
list
that
con-
tains
the
ip-
mi-
tool
com
man
that
was
ex-
e-
cute
it
will

Returns
the
next
ac-
tual

exceptions for user-accessible actions.

Note
that
un-
like
fake
im-
ple-
men-
ta-
tions
thes
do
not
pass
val-
i-
da-
tion
and
raise

class `ir`
Base
obj
Mix
to
add
to
an
in-
ter-
face
to
mak
it
fail
val-
i-
da-
tion.

get_prop

validate

class `ir`
Base
iro
dri

mod
noo
Fai
iro
dri
bas
BIO
BIO
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

apply_co

Val-
i-
date
&
ap-
ply
BIO
set-
tings
on
the
give
node

This
meth
take
the
BIO
set-
tings
from
the
set-

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

Parameters

- **task**
a Task manager instance
- **set_bios**
Dictionary containing the BIOS configuration.

Raises

UnsupportedDriverExtension if the node driver

tion.

does
sup-
port
BIO
con-
fig-
u-
ra-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
set-
tings
fails

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
some
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Returns

state
if
BIO
con-

plete.

fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

cache_b

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing
clea
ing
op-
er-
a-
tion

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

ties from bare metal.

Parameters

task

a

Task

ager

in-

stan-

Raises

Un-

sup-

port

ed-

Driv

ten-

sion

if

the

node

driv

does

sup-

port

get-

ting

BIO

prop

er-

Returns

Non

factory.

Re-

set

BIO

con-

fig-

u-

ra-

tion

to

fac-

tory

de-

fault

on

the

After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

give
node

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node
Af-

Parame
task
a
Task
ager
in-
stan

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
BIO
re-
set.

plete.

Returns
state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

class ir

Base
iro
dri
mod
noo
Fai
iro
dri
bas
Con

Con
sole
in-
ter-
face
im-
ple-
men
ta-
tion
that
raise
er-
rors
on
all
re-

ques

get_conso

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for
the
clie
to
ac-
cess
the

console.

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

start_co

Star
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou
not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read
start

Paramete

task
A
Task
ager
in-
stan-
con-
tain-

ing
the
node
to
act
on.

stop_con

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

class ir

Base
iro
dri
mod
noo
Fai
iro
dri
bas
Ins
In-
spec
in-

ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

inspect_

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tiona
hard
ware
prop
er-
ties.

Parame
task
A
task
from
Task
ager

Raises

Hardware specification failure, if unable to get essential hardware properties.

Returns

Resulting state of the inspection i.e. state or Non

class ir

Base *iron_driver_base.RAI*
RAI interface

im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

create_

Cre-
ates
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
It
as-
sum
that
the

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID config-

uration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

ified in the nodes `target_raid_config`. Default value is True.

Parameters

- **task_manager**
A TaskManager instance.
- **create_non_root_volumes**
Setting this to False indicates not to create root volume that is specified in the nodes `target_raid_config`. Default value is True.
- **create_non_root_volumes**
Setting this to False indicates not to create non-root

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

creating the new configuration.

chronously, or `None` if it is complete.

vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn

delete_c

Dele

ration is deleted, `node.raid_config` should be cleared by the implementation.

RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
Af-
ter
RAI
con-
fig-
u-

Parameter
task
A
Task
ager
in-
stan

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion

it is complete.

this interface can override this method to support custom parameters for RAID configuration.

is
in
prog
asyn
chro
or
Non
if

validate

Val-
i-
date
the
give
RAI
con-
fig-
u-
ra-
tion.

This
meth
val-
i-
date
the
give
RAI
con-
fig-
u-
ra-
tion.
Driv
im-
ple-
men
ta-
tions
of

Paramete

- **task**
A
Task

ager
in-
stan

- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valic

class ir

Base
iro
dri
mod
noo
Fai
iro
dri
bas
Res
Res-
cue

in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

rescue()

Boo
the
task.
node
into
a
res-
cue
en-
vi-
ron-
men

Paramete

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
stan-
cue-

Fail-
ure
if
node
val-
i-
da-
tion
or
res-
cue
op-
er-
a-
tion
fails

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it
is
com
plete

unrescue

Tear
down
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to
nor-
mal.

ation fails.

Parameters

task

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Raises

In-

stan-

Un-

res-

cue-

Fail-

ure

if

node

val-

i-

da-

tion

or

un-

res-

cue

op-

er-

Returns

state

if

it

is

suc-

cess

ful.

class ironic

Base

ironic

driver

module

noop
Fai
iro
dri
bas
Ven
Ven-
dor
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

driver_v

Val-
i-
date
drive
vend
pass
ac-
tions
If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **metl**
meth
to
be
val-
i-
date
- **kwa**
info
for
ac-
tion.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain
cer-
tain
pa-
ram-
e-
ter.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
pa-
ram-
e-
ter

ironic.drivers.modules.noop_mgmt module

does
not
matc

No-
op
man
age-
men
in-
ter-
face
im-
ple-
men
ta-
tion.

class ir

Base
iro
dri
bas
Man

No-
op
man
age-
men
in-
ter-
face
im-
ple-
men
ta-
tion.

Us-
ing
this
im-
ple-
men
ta-
tion
re-

to first try PXE booting, then fall back to hard drives.

quir
the
boot
or-
der
to
be
pre-
con-
fig-
ured

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node
Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa
that
not
all
driv
sup-
port
this.

Paramete

task
A
task
from
Task

ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
iro
comm
boo
or
Non
if
it
is
un-
know

persist

Whe
the

unknown.

boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

get_sens

Get
sen-
sors
data

meth

Parameters

task

A

Task

ager

in-

stan

Raises

Fail

To-

Get-

Sen-

sor-

Data

wh

get-

ting

the

sen-

sor

data

fails

Raises

Fail

ToP

eSer

sor-

Data

wh

pars

ing

sen-

sor

data

fails

Returns

Re-

turn

a

con-

sis-

tent

for-

mat

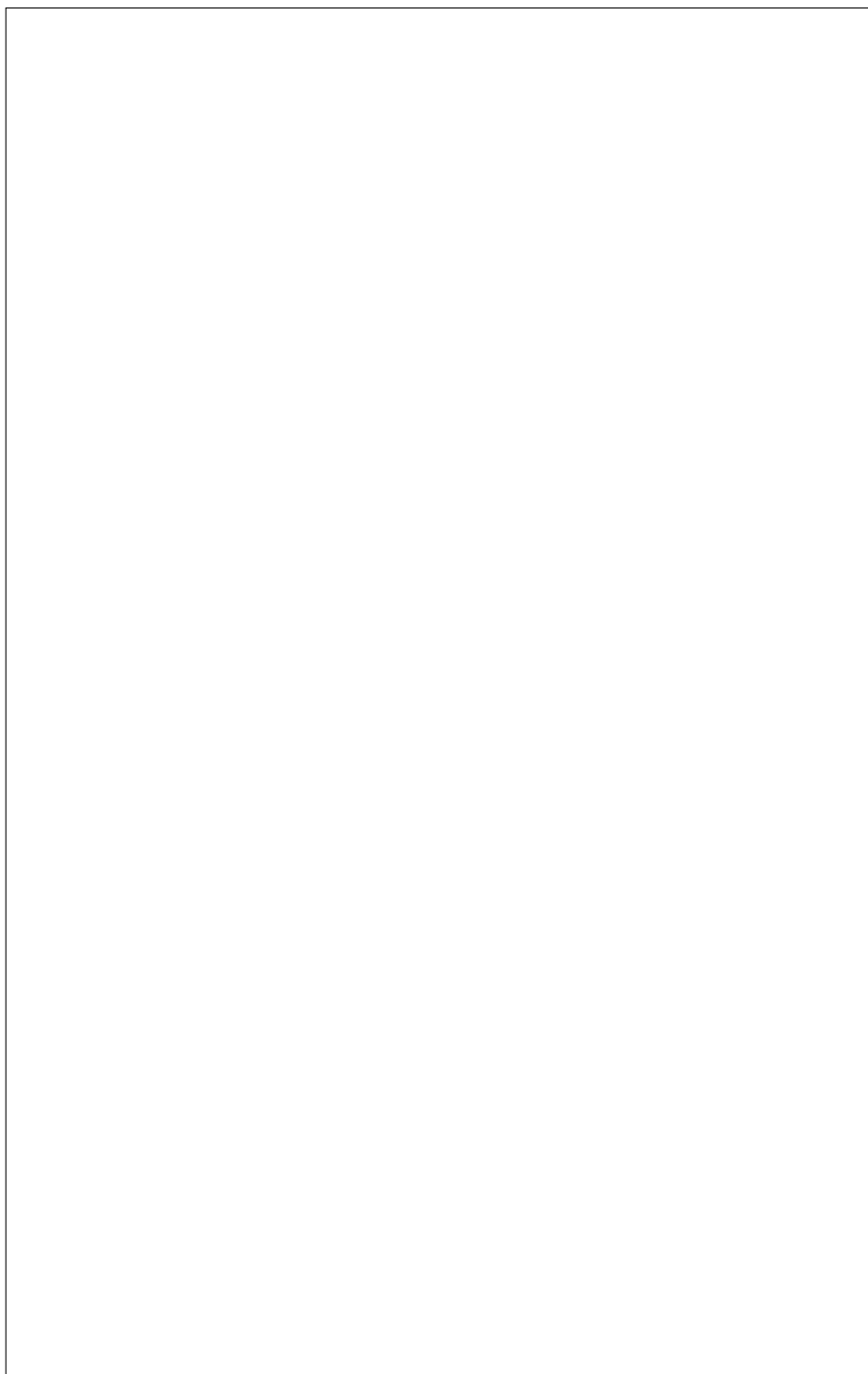
dict

of

sen-

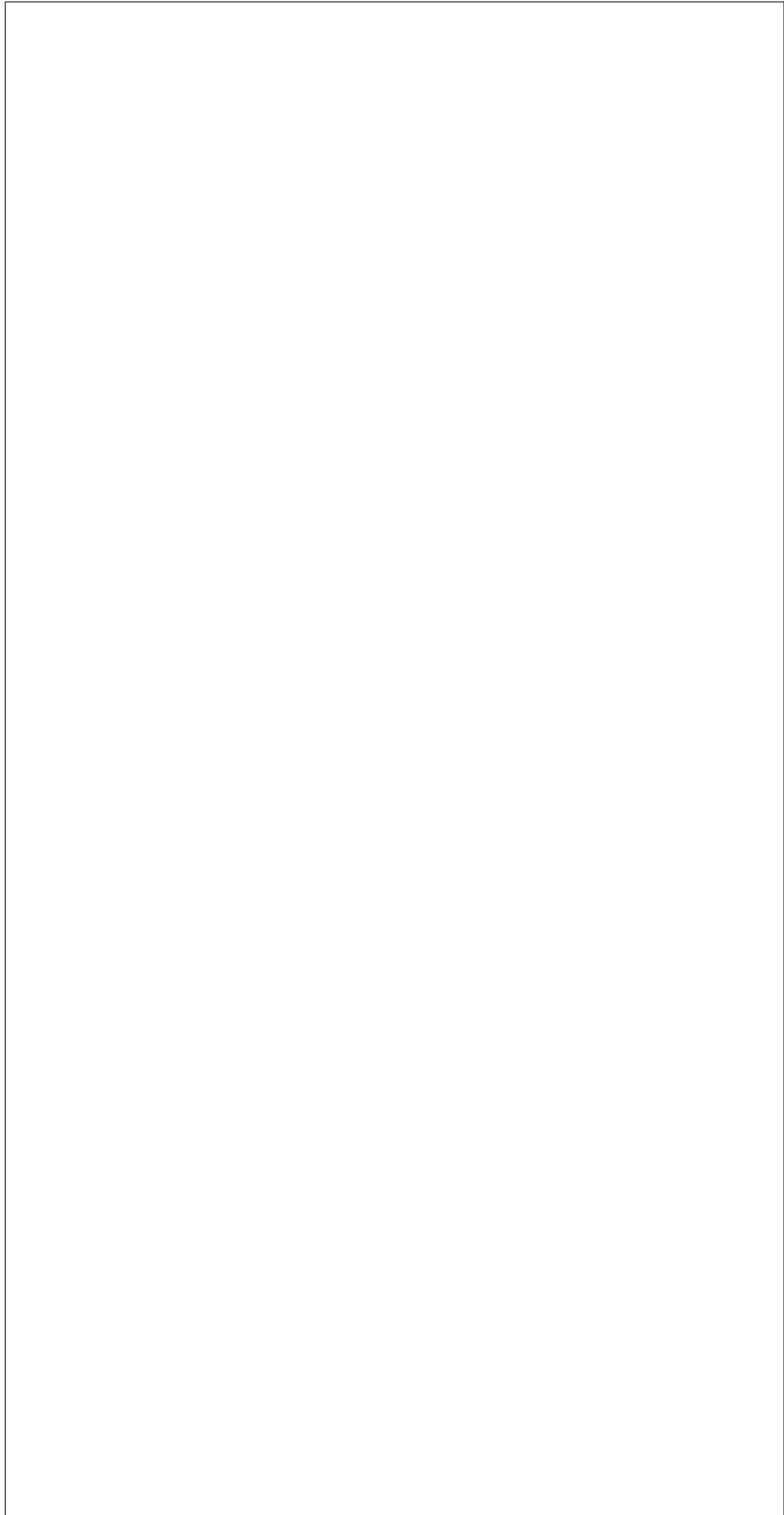
sor
data
grou
by
sen-
sor
type
whic
can

be processed by Ceilometer. eg,



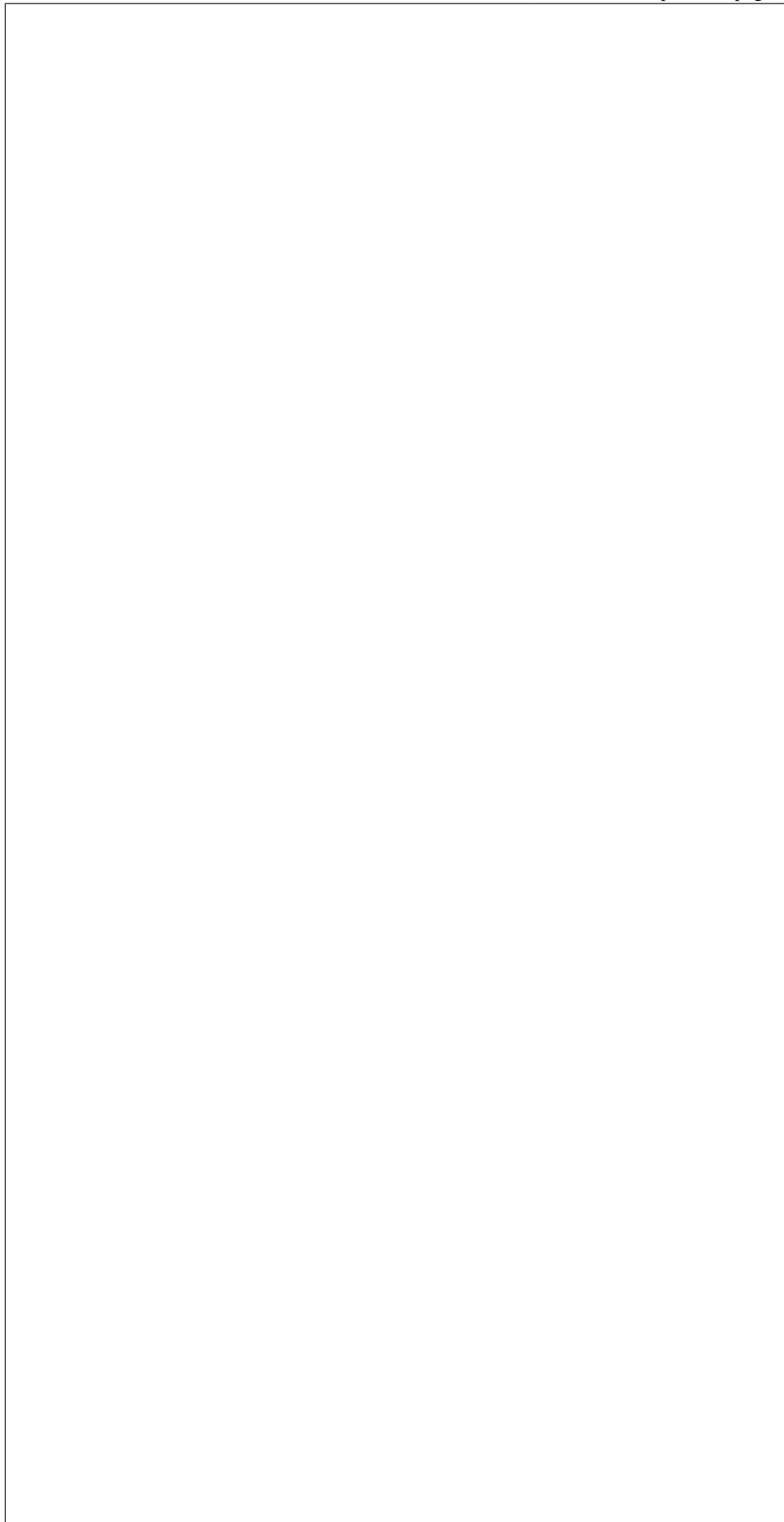
(continues on next page)

(continued from previous page)



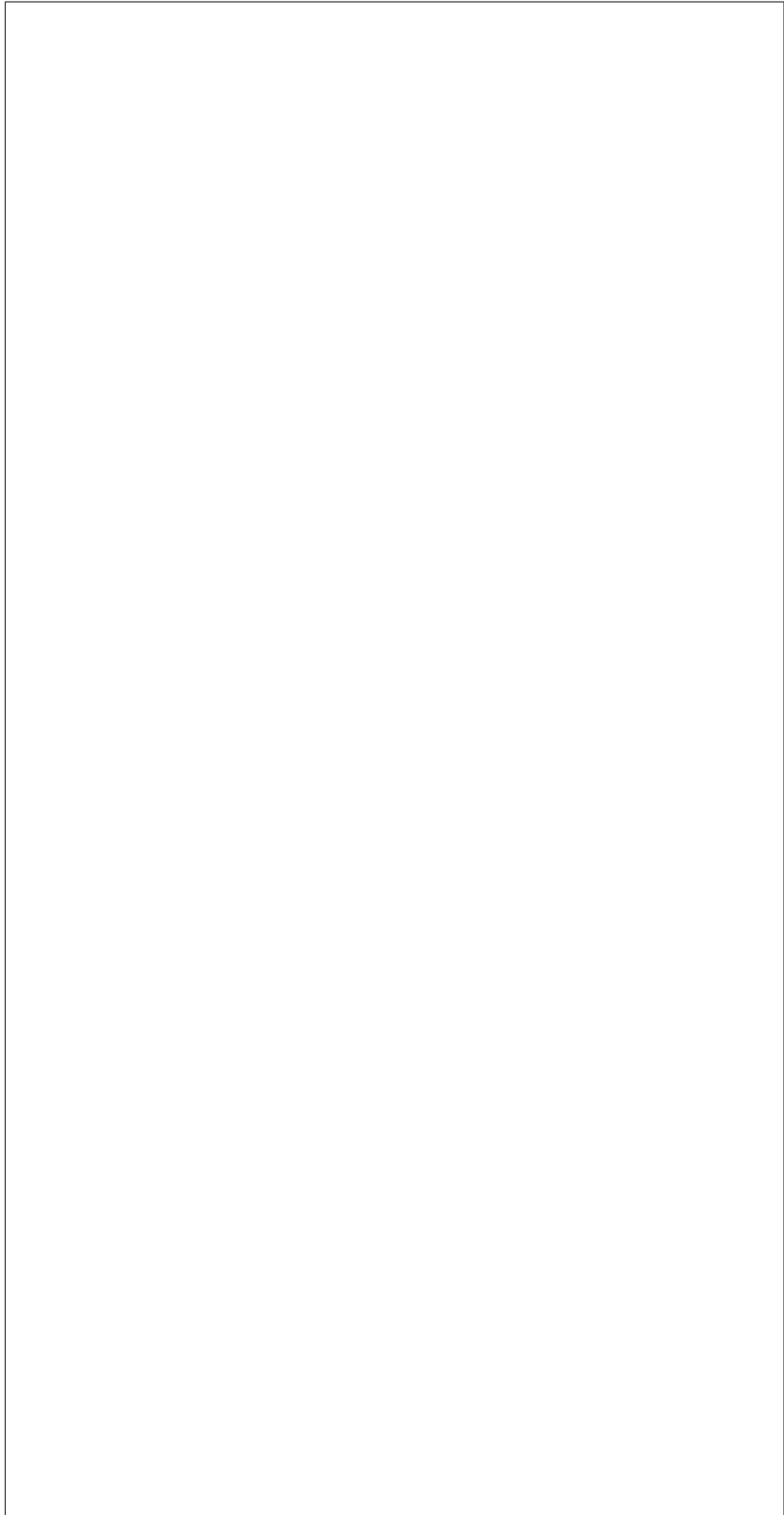
(continues on next page)

(continued from previous page)



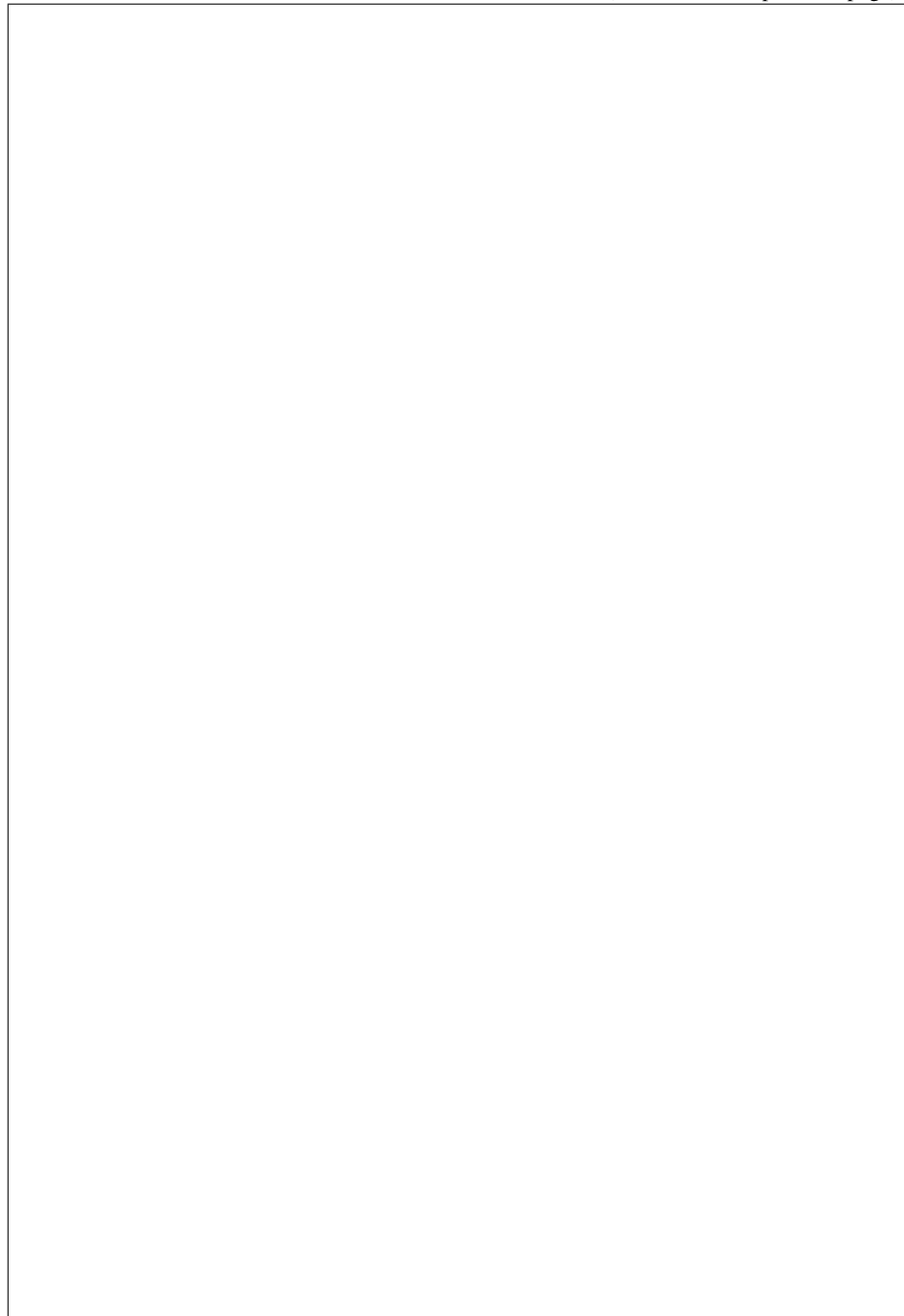
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



get_supp

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter

task

A task from Taskager

Returns

A list with the supported boot device defined in [ironic-compute-
bootstrap](#)

set_boot

Set the boot device for a node

Set the boot device to use on next reboot of the node

Parameter

not. Default: False.

- **task**
A
task
from
Task
ager

- **dev.**
The
boot
de-
vice
one
of
ironiC
comm
boo

- **per.**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-

valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.
This
meth
val-
i-
date
whe
the
driv
and/
in-

the required information for this interface to function.

long-running checks.

stand
prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid

Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s

ironic.drivers.modules.pxe module

PXE
Boo
In-
ter-
face

class ir

Base
iro
dri
mod
age
Age
iro
dri
mod
age
Hea
iro

dri
bas
Dep

deploy()

Per-
form
a
de-
ploy
men
to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node

to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

deploy_1

deploy_2

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
de-

this method should be implemented by the driver.

ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahea
of
time
is
pos-
si-
ble,

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time

the same node on the same conductor.

for

This
meth
is
call
be-
fore
*de-
ploy*

Parame

task

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

reboot_t

Met
in-
voke
af-
ter
the
de-
ploy
men
is
com
plete

Parame

task

a
Task
ager
in-
stan

should_r

Whe

agen
boot
is
man
aged
by
iron

validate

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-

long-running checks.

chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
mod
pxe
PXE
iro
dri
bas
Boo

capabil:

class ir

Base
iro
dri
mod
age
Age
iro
dri
mod
age
Hea
iro
dri
bas
Dep

deploy(t

Per-
form
a
de-
ploy
men
to
the

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto
the
spec-
i-
fied
node
This

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy

this method should be implemented by the driver.

the same node on the same conductor.

men
en-
vi-
ron-
men-
ahead
of
time
is
pos-
si-
ble,

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
*de-
ploy*

Paramet
task
A
Task
ager
in-
stan
con-

tain-
ing
the
node
to
act
on.

validate

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-

long-running checks.

chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

`ironic.drivers.modules.pxe_base` module

on
miss
ing
pa-
ram-
e-
ter(s)

Base
PXE
In-
ter-
face
Met
ods

class `ir`
Base
obj

clean_up

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

links the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

It
un-

Parameters
task
a
task
from
Task
ager

Returns
Non

clean_up

Clea
up
the
boot
of
iron
ram

This
meth
clea
up
the
PXE
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy
or

rescue ramdisk. It unlinks the deploy/rescue kernel/ramdisk in the nodes directory in tftproot and removes its PXE config.

Parameters

- **task**
a

ried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation was carried out.

task
from
Task
ager

•
mod
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue
op-
er-
a-
tion
was
car-

Returns
Non

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam

de-
scrip-
tion:
en-
tries

ipxe_ena

prepare_

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config.

Parame

tasl
a
task
from
Task
ager

Returns

Non

prepare_

evant information from the nodes driver_info and instance_info.

Pre-
pare
the
boot
of
Iron
ram
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ker-
nel/
af-
ter
read
ing
rel-

Parameters

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to

ters as kernel command-line arguments.

or instance_info.

be
pass
to
the
ram
pxe
driv
pass
thes
pa-
ram-
e-

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
driv

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

the node.

pro-
vide
is
in-
valid

Raises

Iron
icEx
cep-
tion,
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on

validate

Val-
i-
date
the
PXE
spec
info
for
boot
ing
de-
ploy
im-
ages

This
meth
val-
i-
date
the
PXE
spec

If invalid, raises an exception; otherwise returns None.

info
for
boot
ing
the
ram
and
in-
stan
on
the
node

Parameter
task
a
task
from
Task
ager

Returns
Non

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
in-
valid

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som

re-
quir
pa-
ram-
e-
ters
are
miss
ing.

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parameter

task
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
res-
cue.

Parame

task
a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram
e-
ter-
Valu
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-

eters

ironic.drivers.modules.snmp module

driver model to support devices with different SNMP object models.

Iron
SNM
pow
man
ager

Pro-
vide
ba-
sic
pow
con-
trol
us-
ing
an
SNM
enab
sma
pow
con-
troll
Use
a
plug
gabl

class ir

Base
obj

SNM
clien
ob-
ject.

Per-
form
low
leve
SNM
get
and
set

tion with PySNMP to simplify dynamic importing and unit testing.

op-
er-
a-
tion:
En-
cap-
su-
lates
all
in-
ter-
ac-

get (*oid*)

Use
PyS
NM
to
per-
form
an
SNM
GET
op-
er-
a-
tion
on
a
sin-
gle
ob-
ject.

Parame

oid
The
OID
of
the
ob-
ject
to
get.

Raises

SN-
MP-
Fail-
ure
if

an
SNM
re-
ques
fails

Returns

The
valu
of
the
re-
ques
ob-
ject.

get_next

Use
PyS
NM
to
per-
form
an
SNM
GET
NEX
op-
er-
a-
tion
on
a
ta-
ble
ob-
ject.

Param

oid
The
OID
of
the
ob-
ject
to
get.

Raises

SN-
MP-

Fail-
ure
if
an
SNM
re-
ques
fails

Returns

A
list
of
val-
ues
of
the
re-
ques
ta-
ble
ob-
ject.

set(*oid*,

Use
PyS
NM
to
per-
form
an
SNM
SET
op-
er-
a-
tion
on
a
sin-
gle
ob-
ject.

Parame

- **oid**
The
OID

of
the
ob-
ject
to
set.

- **valu**
The
valu
of
the
ob-
ject
to
set.

Raises
SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

class ir

Base
iro
dri
mod
snm
SNM

SNM
driv
class
for
APC
Mas
ter-
Swit
PDU
de-
vice

SNM
ob-
jects

ues: 1=On, 2=Off, 3=PowerCycle, [more options follow]

for
APC
SN-
M-
P-
Driv
APC
Mas
ter-
Swit
PDU
1.3.0
sP-
DU-
Out-
letC
Val-

oid_dev:

system_i:

value_p:

value_p:

class ir

Base
iro
dri
mod
snmp
SNM
SNM
driv
class
for
APC
Mas
ter-
Swit
Plus
PDU
de-
vice
SNM
ob-

ControlMSPOutletCommand Values: 1=On, 3=Off, [more options follow]

jects
for
APC
SN-
M-
P-
Driv
APC
Mas
ter-
Swit
Plus
PDU
1.3.6
sP-
DU-
Out-
let-

oid_dev:

system_:

value_po

value_po

class ir

Base
iro
dri
mod
snm
SNM

SNM
driv
class
for
APC
Rack
PDU
de-
vice

SNM
ob-
jects
for

letCommand Values: 1=On, 2=Off, 3=PowerCycle, [more options follow]

APC
SN-
M-
P-
Driv
APC
PDU
PDU

1.3.0
rP-
DU-
Out-
let-
Con
trolC

oid_dev:

system_i:

value_p:

value_p:

class ir

Base
iron
dri
mod
snmp
SNM

SNM
driv
class
for
Ater
PDU
de-
vice

SNM
ob-
jects
for
Ater
PDU
1.3.0

Out-
let
Pow
Val-
ues:
1=O
2=O
3=P
ing,
4=R
set

oid_dev:

system_:

value_po

value_po

class ir

Base
iron
drive
mod
snmp
SNM

SYS_OBJ.

class ir

Base
obj
SNM
pow
drive
base
class
The
SN-
M-
P-
Drive
class
hi-

to interface with different smart power controller products.

er-
ar-
chy
im-
ple-
men
man
spec
MIE
ac-
tions
over
SNM

oid_ente

power_on

Set
the
pow
state
to
this
node
to
OFF

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
ironic
comm
sta

power_on

Set

the
pow
state
to
this
node
to
ON.

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
iro
com
sta

power_re

Re-
set
the
pow
to
this
node

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow

state
One
of
iro
com
sta

power_st

Re-
turn
a
node
cur-
rent
pow
state

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
iro
com
sta

retry_in

class ir

Base
iro
dri
mod
snm
SNM

SNM
driv
class
for

0=Off, 1=On, 2=Reboot

Bay
MRI
PDU
de-
vice
SNM
ob-
jects
for
Bay
MRI
PDU
4775
1,
3,
5,
3,
1,
3,
{uni
Out-
let
Pow
Val-
ues:

oid_dev:

unit_id

value_p

value_p

class ir

Base
iro
dri
mod
snm
SNM
SNM
drive
class
for
Cy-
ber-

1=On, 2=Off, 3=PowerCycle, [more options follow]

Pow
PDU
de-
vice

SNM
ob-
jects
for
Cy-
ber-
Pow
PDU
1.3.0
eP-
DU-
Out-
let-
Con
trolC
let-
Com
man
Val-
ues:

oid_dev:

system_i

value_p

value_p

class ir

Base
ironic
driver
module
snmp
SNMP

SNM
driv
class
for
Eato
Pow
PDU

multiple SNMP objects.

ing off, 3=pending on 1.3.6.1.4.1.534.6.6.7.6.6.1.3.<outlet ID> outletControlOffCmd Write 0 for immediate power off 1.3.6.1.4.1.534.6.6.7.6.6.1.4.<outlet ID> outletControlOnCmd Write 0 for immediate power on

The
Eato
pow
PDU
does
not
fol-
low
the
mod
of
SN-
M-
P-
Driv
Sim
ple
as
it
uses

SNM
ob-
jects
for
Eato
Pow
PDU
1.3.6.
let
ID>
out-
let-
Con
trol-
Sta-
tus
Reac
0=on
1=on
2=pe

oid_dev:

oid_pow

oid_pow

oid_stat

status_c

status_c

status_p

status_p

system_f

value_p

value_p

class ir

Base
iro
dri
mod
snm
SNM

SNM
drive
base
class
for
sim-
ple
PDU
de-
vice

Here
sim-
ple
refer
to
de-
vice
whic
pro-
vide
a

the power state of an outlet.

A different OID may be specified by overriding the `_snmp_oid` method in a subclass.

sin-
gle
SNM
ob-
ject
for
con-
trol-
ling

The
de-
fault
OID
of
the
pow
state
ob-
ject
is
of
the
form
<en-
ter-
prise
OID
OID
ID>

abstract

De-
vice
de-
pen-
dent
por-
tion
of
the
pow
state
ob-
ject
OID

abstract

Valu
rep-
re-

sent
ing
pow
off
state

abstract

Valu
rep-
re-
sent
ing
pow
on
state

class ir

Base
iron
dri
mod
snmp
SNM

SNM
driv
class:
for
Tel-
tron
PDU
de-
vice

SNM
ob-
jects
for
Tel-
tron
PDU
1.3.6
Out-
let
Pow
Val-
ues:
1=O
2=O

oid_dev:

system_

value_p

value_p

class ir

Base
iro
dri
bas
Pow

SNM
Pow
In-
ter-
face

This
Pow
er-
In-
ter-
face

class
pro-
vide
a
mec
a-
nism
for
con-
trol-
ling
the
pow
state

of a physical device using an SNMP-enabled smart power controller.

get_powe

Get
the
cur-
rent
pow
state

Poll

the
SNM
de-
vice
for
the
cur-
rent
pow
state
of
the
node

Parameter
task

An
in-
stan-
ce
of
iron

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
re-
quir-
SNM
pa-
ram-
e-
ters
are
miss-
ing.

Raises

In-
valid-
Pa-
ram-
e-
ter-
Valu-
if
SNM
pa-

ram-
e-
ters
are
in-
valid

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
ironic
comm
sta

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:

en-tries

reboot (*n*)

Cycles the power to a node

Parameters

- **task**
An instance of *iron*

- **time**
time out (in seconds)
Unsupported by this interface

Raises

MissingParameterError
ValueError
if required SNM parameter

e-
ters
are
miss
ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
SNM
pa-
ram-
e-
ters
are
in-
valid

Raises

Pow
er-
State
Fail-
ure
if
the
fi-
nal
pow
state
of
the
node
is
not
POV
af-
ter

the timeout.

Raises

SN-
MP-
Fail-
ure
if
an

SNM
re-
ques
fails

set_power

Turn
the
power
on
or
off.

Set
the
power
state
of
a
node

Parameters

- **task**
An
in-
stance
of
iron
- **power**
Either
POWER
or
POWER
from
:class
iron
- **timeout**
time
out
(in
sec-
onds
Un-
sup-
port

by
this
in-
ter-
face

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
re-
quir-
SNM
pa-
ram-
e-
ters
are
miss-
ing.

Raises

In-
valid-
Pa-
ram-
e-
ter-
Valu-
if
SNM
pa-
ram-
e-
ters
are
in-
valid
or
psta
is

invalid.

Raises

Pow-
er-
State
Fail-

after the timeout.

ure
if
the
fi-
nal
pow
state
of
the
node
is
not
as
re-
ques

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

validate

Che
that
node
con-
tain
the
req-
ui-
site
field

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir

SNM
pa-
ram-
e-
ters
are
miss
ing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
SNM
pa-
ram-
e-
ters
are
in-
valid

`ironic.d`

`ironic.d`

Module contents

Submodules

`ironic.drivers.base` module

Ab-
strac
base
class
for
driv

`ironic.d`

`'inspect'`

`'vendor'`

Con
stan

hold
ing
all
know
in-
ter-
face

class ir

Base
iro
dri
bas
Bas

abstract

Val-
i-
date
&
ap-
ply
BIO
set-
tings
on
the
give
node

This
meth
take
the
BIO
set-
tings
from
the
set-
tings
para
and
ap-
plies
BIO
set-
tings
on
the

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

Parameters

- **task**
a Task manager instance
- **set**
Dictionary containing the BIOS configuration.

Raises

UnsupportedDriverException if the node driver does not support BIOS configuration.

tion.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of
set-
ting
fails

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or

plete.

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

Non
if
it
is
com

abstract

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing
clea
ing
op-
er-
a-
tion

Paramet

task
a
Task
ager
in-
stan

ties from bare metal.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
get-
ting
BIO
prop
er-

Returns

Non

abstract

Re-
set
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node

This
meth
re-
sets
BIO
con-
fig-
u-

ter the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

ra-
tion
to
fac-
tory
de-
fault
on
the
give
node
Af-

Parameters

task
a
Task
ager
in-
stan-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-
port
BIO
re-
set.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in

plete.

prog
asyn
chro
or
Non
if
it
is
com

interfa

In-
ter-
face
type
used
for
clear
step
and
log-
ging

class ir

Base
obj

A
bare
driv
ob-
ject
whic
will
have
in-
ter-
face
at-
tach
later

Any
com
pos-
able
in-
ter-
face
shou
be
adde

appended to `core_interfaces` or `standard_interfaces` here.

as
class
at-
tribu
of
this
class
as
well
as

property

bios = BIOS

Standard data attribute for BIOS related features.
A reference to an instance of `BIOS` class.

boot = BOOT

Standard data attribute for boot related features.
A reference to an instance of `BOOT` class.

to
an
in-
stan
of
:clas

console

Stan
dara
at-
tribu
for
man
ag-
ing
con-
sole
ac-
cess

A
ref-
er-
ence
to
an
in-
stan
of
:clas

property

In-
ter-
face
that
are
re-
quir
to
be
im-
ple-
men

deploy =

Cor
at-
tribu
for
man
ag-

ing
de-
ploy
men

A
ref-
er-
ence
to
an
in-
stan
of
:clas

get_prop

Get
the
prop
er-
ties
of
the
driv

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Stan
dard
at-
tribu
for
in-
spec
tion
re-
latec
fea-

tures
A
ref-
er-
ence
to
an
in-
stan-
of
:clas

managem

Stan
dara
at-
tribu
for
man
age-
men
re-
latec
fea-
tures

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

network

Cor
at-
tribu
for
net-
worl
con-
nec-
tiv-
ity.

A
ref-
er-
ence

to
an
in-
stan-
of
:clas

property

property

In-
ter-
face
that
can
be
no-
op.

power =

Cor
at-
tribu
for
man
ag-
ing
pow
state

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

raid = 1

Stan
dara
at-
tribu
for
RAI
re-
latec
fea-
ture

A
ref-
er-
ence
to
an
in-
stan-
ce
of
:clas

rescue =

*Stan-
dard*
at-
tribu-
tion
for
ac-
cess-
ing
res-
cue
fea-
tures

A
ref-
er-
ence
to
an
in-
stan-
ce
of
:clas

storage

*Stan-
dard*
at-
tribu-
tion
for
(re-
mot-
stor-
age
in-
ter-
face

A
ref-
er-

ence
to
an
in-
stan
of
:clas

vendor =

At-
tribu
for
ac-
cess
ing
any
vend
spec
ex-
ten-
sion

A
ref-
er-
ence
to
an
in-
stan
of
:clas

class ir

Base
obj

A
base
in-
ter-
face
im-
ple-
men
ing
com
mon
func
tions
for
Driv

In-
ter-
face

execute_

Ex-
e-
cute
the
clear
step
on
task

A
clear
step
must
take
a
sin-
gle
po-
si-
tion.
ar-
gu-
men-
a
Task
ager
ob-
ject.
It

may take one or more keyword variable arguments (for use with manual cleaning only.)

A
step
can
be
ex-
e-
cute
syn-
chro-
or
asyn-
chro-
A
step
shou

method has completed synchronously or states.CLEANWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the continue_node_clean RPC, so the conductor can begin the next clean step.

re-
turn
Non
if
the

Parame

- **task**
A
Task
ager
ob-
ject
- **step**
The
clea
step
dic-
tio-
nary
rep-
re-
sent
ing
the
step
to
ex-
e-
cute

Returns

Non
if
this
meth
has
com
plete
syn-
chro
or
state
if
the

ecute asynchronously.

It may take one or more keyword variable arguments (for use in the future, when deploy steps can be specified via the API).

step
will
con-
tinue
to
ex-

execute,

Ex-
e-
cute
the
de-
ploy
step
on
task

A
de-
ploy
step
must
take
a
sin-
gle
po-
si-
tion.
ar-
gu-
men-
a
Task
ager
ob-
ject.

A
step
can
be
ex-
e-
cute
syn-
chro-
or

method has completed synchronously or states.DEPLOYWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the continue_node_deploy RPC, so the conductor can begin the next deploy step.

Parameters

- **task**
A Task object
- **step**
The deployment step dictionary representing the step to execute

Returns

None if this method has completed

ecute asynchronously.

syn-
chro-
or
state
if
the
step
will
con-
tinue
to
ex-

get_clea

Get
a
list
of
(en-
able
and
dis-
able
clea
step
for
the
in-
ter-
face

This
func
tion
will
re-
turn
all
clea
step
(bot
en-
able
and
dis-
able
for
the
in-
ter-

in an unordered list.

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available clean steps.

face

Parameters

tasks

A
Task
ager
ob-
ject,
use-
ful
for
in-
ter-
face
over
rid-
ing
this
func-
tion

Raises

NodeCleanStepsError

if
there
is
a
prob-
lem
get-
ting
the
steps
from
the
driver
For
ex-
am-
ple,
when

Returns

A
list
of
clean
step
dic-

tio-
nar-
ies

get_dep

Get
a
list
of
(en-
able
and
dis-
able
de-
ploy
step
for
the
in-
ter-
face

This
func-
tion
will
re-
turn
all
de-
ploy
step
(bot
en-
able
and
dis-
able
for
the
in-
ter-

face, in an unordered list.

Param

task
A
Task
ager
ob-
ject,

use-
ful
for
in-
ter-
face
over
rid-
ing
this
func-
tion

Raises

Ins
if
there
is
a
prob-
lem
get-
ting
the
step
from
the
drive
For
ex-
am-
ple,
whe

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available deploy steps.

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-
ies

abstract

Re-
turn
the

prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

In-
ter-
face
type
used
for
clear
steps
and
log-
ging

support

In-
di-
cate
if
an
in-
ter-
face
is
sup-
port

This
will
be
set

or in the process of being deprecated.

to
Fals
for
in-
ter-
face
whic
are
unte
in
first-

or
thir
part
CI,

abstract

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info.

This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

the required information for this interface to function.

long-running checks.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-

e-
ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-
ter(s)

class ir

Base
iro
dri
bas
Bas
In-
ter-
face
for
boot
relat
ac-
tion

capabil:

abstract

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the

en-
vi-
ron-
men-
that
was
setu
for
boot
ing
the
in-
stan

Parameter

task
A
task
from
Task
ager

Returns

Non

abstract

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy

cue ramdisk.

or
res-

Parame
task
A
task
from
Task
ager

Returns
Non

interfa
In-
ter-
face
type
used
for
clear
step
and
log-
ging

abstract

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-

tion from the nodes database.

vant information from the nodes database.

vant
in-
for-
ma-

Parame

tas

A
task
from
Task
ager

Returns

Non

abstract

Pre-
pare
the
boot
of
Iron
rame

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
rame
af-
ter
read
ing
rel-
e-

Parame

•
tas

might want to boot the ramdisk in different ways by passing parameters to them. For example,

etc.

A
task
from
Task
ager

- **ram**
The
op-
tions
to
be
pass
to
the
iron
ram
Dif-
fer-
ent
im-
ple-
men
ta-
tions

Whe
Age
ram
is
boot
to
de-
ploy
a
node
it
take
the
pa-
ram-
e-
ters
ipa-
api-
url,

Othe
im-
ple-

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

men
ta-
tions
can
mak
use
of
ram
to
pass
such
in-
for-
ma-
tion.
Dif-
fer-

Returns

Non

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Paramet

task
A
Task
ager
in-
stan
with
the
node
be-
ing

eters

che
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-

Raises
Un-
sup-
port
ed-
Driv
ten-
sion

validate
Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
res-
cue.

**Paramet
task**

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-

eters

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

class ir

Base
iro
dri
bas
Bas
In-
ter-

face
for
cons
relat
ac-
tions

abstract

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for
the
clie
to
ac-
cess
the

console.

Parame

task
A
Task
ager
in-
stan-
con-

tain-
ing
the
node
to
act
on.

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

interface

In-
ter-
face
type
used
for
clear
steps
and
log-
ging

abstract

Star
a
re-
mot
con-
sole
for
the
task
node

This
meth
shou
not
raise
an
ex-

cep-
tion
if
con-
sole
al-
read
start

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

abstract

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

act
on.

class ir

Base
iron
dri
bas
Bas

In-
ter-
face
for
depl
relat
ac-
tion

abstract

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

is
pos-
si-
ble,

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
tear

Paramet

task

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

abstract

Per-
form
a
de-
ploy
men
to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This

method will be called after prepare(), which may have already performed any preparatory steps, such as pre-caching some data for the node.

Parame

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

heartbeat

Rece
a
hear
beat
for
the
node

Parameter

- **task**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **cal**
a
URI
to
use
to
call
to
the
ram
- **age**
The

ver-
sion
of
the
ager
that
is
hear
beat
ing

- **ager**
TLS
cer-
tifi-
cate
for
the
ager

- **ager**
Sta-
tus
of
the
hear
beat
ing
ager

- **ager**
Mes-
sage
de-
scrib-
ing
the
ager
sta-
tus

Returns
Non

interface
In-
ter-
face
type
used
for

clear
steps
and
log-
ging

abstract

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

this method should be implemented by the driver.

If
im-
ple-
men
this
meth
mus
be

the same node on the same conductor.

iden
po-
tent.
It
may
be
called
mul-
ti-
ple
time
for

This
meth
is
called
be-
fore
*de-
ploy*

Parame

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

prepare

Pre-
pare
the
node
for
clear-
ing
task.
For
ex-
am-

to do in-band cleaning tasks.

they would be set in `ironic.conductor.manager._do_node_clean`, but cannot be set when this is asynchronous. After, the interface should make an RPC call to `continue_node_cleaning` to start cleaning.

ple,
node
that
use
the
Iron
Pyth
Age
will
need
to
boot
the
rame
in
or-
der

If
the
func
tion
is
asyn
chro
the
driv
will
need
to
han-
dle
set-
tings
node
and
node
as

NOT
this
shou
be
mov
to
Boo
In-
ter-
face

when
it
gets
im-
ple-
men

Parameters

task

A
Task
object
representing
an
instance
of
a
task
to
be
performed
on
the
node.
The
task
must
be
in
the
state
of
being
ready
to
execute.

Returns

If
this
function
is
successful,
it
returns
a
tuple
containing
the
task
object
and
the
state
of
the
node.
Otherwise,
it
returns
None.

return *None*. The interface will need to call `_get_cleaning_steps` and then RPC to `continue_node_cleaning`

abstract

Take
over
man-
age-
men

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

of
this
task
node
from
a
deac
con-
duc-
tor.
If
con-
duc-
tors
host
main
tain
a
stati
re-
la-
tion-
ship
to
node
this
meth
shou
be
im-

For exam

Neu
tron
mus
for-
war
DH
BO
re-
ques
to
a
con-
duc-
tor
whic
has
pre-

tftpboot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

pare
the

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

abstract

Tear
dow
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node
Give
a
node
that
has
been
pre-
vi-
ousl
de-
ploy
to,
do
all

sary to un-deploy that node.

clear
and
tear
down
nec-
es-

Parameters

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

tear_down

Tear
down
af-
ter
clear
ing
is
com-
plete
Give
that
clear
ing
is
com

node to be deployed to again.

plete
do
all
clea
and
tear
dow
nec-
es-
sary
to
al-
low
the

NOT
this
shou
be
mov
to
Boo
In-
ter-
face
whe
it
gets
im-
ple-
men

Paramete

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

class ir

Base
iron

dri
bas
Bas
In-
ter-
face
for
insp
relat
ac-
tion

ESSENTIAL

The
prop
er-
ties
re-
quir
by
sche
uler

abort (*ta*)

Ab
asyn
chro
nize
hard
ware
in-
spec
tion.

Ab
an
on-
go-
ing
hard
ware
in-
tro-
spec
tion.
this
is
only
used
for

inspect interface.

mentation is expected to be a quick processing.

asyn
chro
nize
base

NOT
This
in-
ter-
face
is
calle
with
node
ex-
clu-
sive
lock
held
the
in-
ter-
face
im-
ple-

Parame
task
a
task
from
Task
ager

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
meth
is
not
im-
ple-
men
by

spect interface.

spe-
cific
in-

abstract

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tiona
hard
ware
prop
er-
ties.

Parame

task
A
task
from
Task
ager

Raises

Har
ware
spec
tion-
Fail-
ure,
if
un-
able
to
get

es-
sen-
tial
hard
ware
prop
er-
ties.

Returns

Re-
sult-
ing
state
of
the
in-
spec
tion
i.e.
state
or
Non

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

class ir

Base
iron
dri
bas
Bas

In-
ter-
face
for
man
age-
men
re-

latec
ac-
tions

detect_v

De-
tect
store
and
re-
turn
the
hard
ware
ven-
dor.

If
the
Node
ob-
ject
prop
field
does
not
al-
read
con-
tain
a
ven
field
then
this
meth
is

intended to query Detects the BMC hardware vendor and stores the returned value with-in the Node object properties field if detected.

Paramet

task
A
task
from
Task
ager

Raises

In-
valid

state is specified.

Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
in-
di-
ca-
tor
or

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Strin
rep-
re-
sent
ing
the
BM
re-
port
Ven-
dor
or

returns None.

Man
u-
fac-
ture
oth-
er-
wise

abstract

Get
the
cur-
rent
boot
de-
vice
for
a
node

Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa
that
not
all
driv
sup-
port
this.

Paramet

task
A
task
from
Task
ager

Raises

Miss

ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
iron
comm
boo
or
Non
if
it
is
un-
know

persist

Whe
the
boot
de-
vice

unknown.

will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_boot

Get
the
cur-
rent
boot
mod
for
a
node

Pro-
vide
the
cur-
rent
boot
mod
of
the
node

NOTE:

may
not
im-
ple-
men
that.

Param

task
A
task
from
Task

ager

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Driv
Op-
er-
a-
tion
or
its
deri
tive
in
case
of
driv
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
re-
ques
op-

the driver

er-
a-
tion
is
not
sup-
port
by

Returns

The
boot
mod
one
of
iron
com
boo
or
Non
if
it
is
un-
know

get_ind:

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard
ware
com
po-
nent

Paramet

- **task**
A

task
from
Task
ager

- **comp**
The
hard
ware
com
po-
nent
one
of
iron
com
com

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
or
in-
di-
ca-
tor

is specified.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Cur-
rent
state
of
the
in-
di-
ca-
tor,
one
of
iro
com
ind

get_mac_

Get
MA
ad-
dres
in-
for-
ma-
tion
for
the
node

Parame

tas
A

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion

Returns

A
list
of
MA
ad-
dres
for
the
node

get_secu

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

NOTE:

may
not
im-
ple-
men

that.

Parameter

task

A

task

from

Task

ager

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

if

a

re-

quir

pa-

ram-

e-

ter

is

miss

ing

Raises

Driv

Op-

er-

a-

tionl

or

its

deri

tive

in

case

of

driv

run-

time

er-

ror.

Raises

Un-

sup-

port

the hardware

ed-
Driv
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
driv
or

Returns

Boo

abstract

Get
sen-
sors
data
meth

Paramet

task

A
Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data
fails

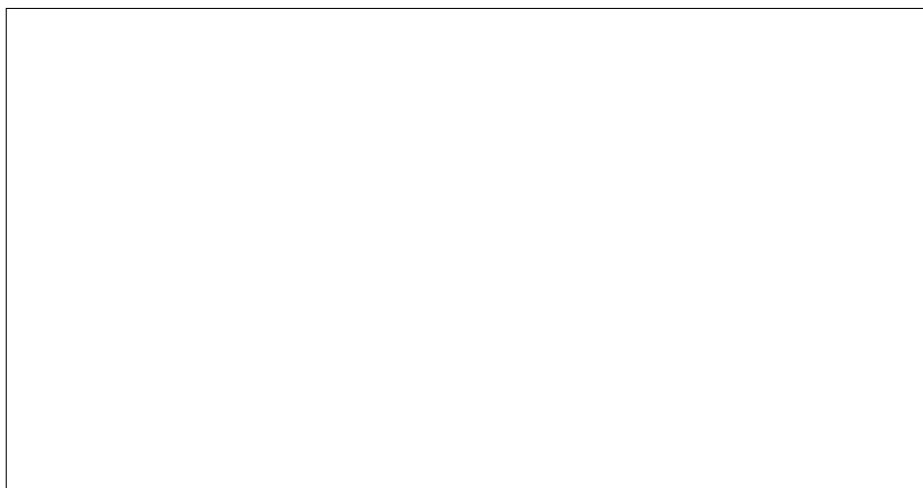
Raises

Fail-
ToP-
eSer-
sor-
Data-
whe
pars
ing
sen-
sor
data
fails

Returns

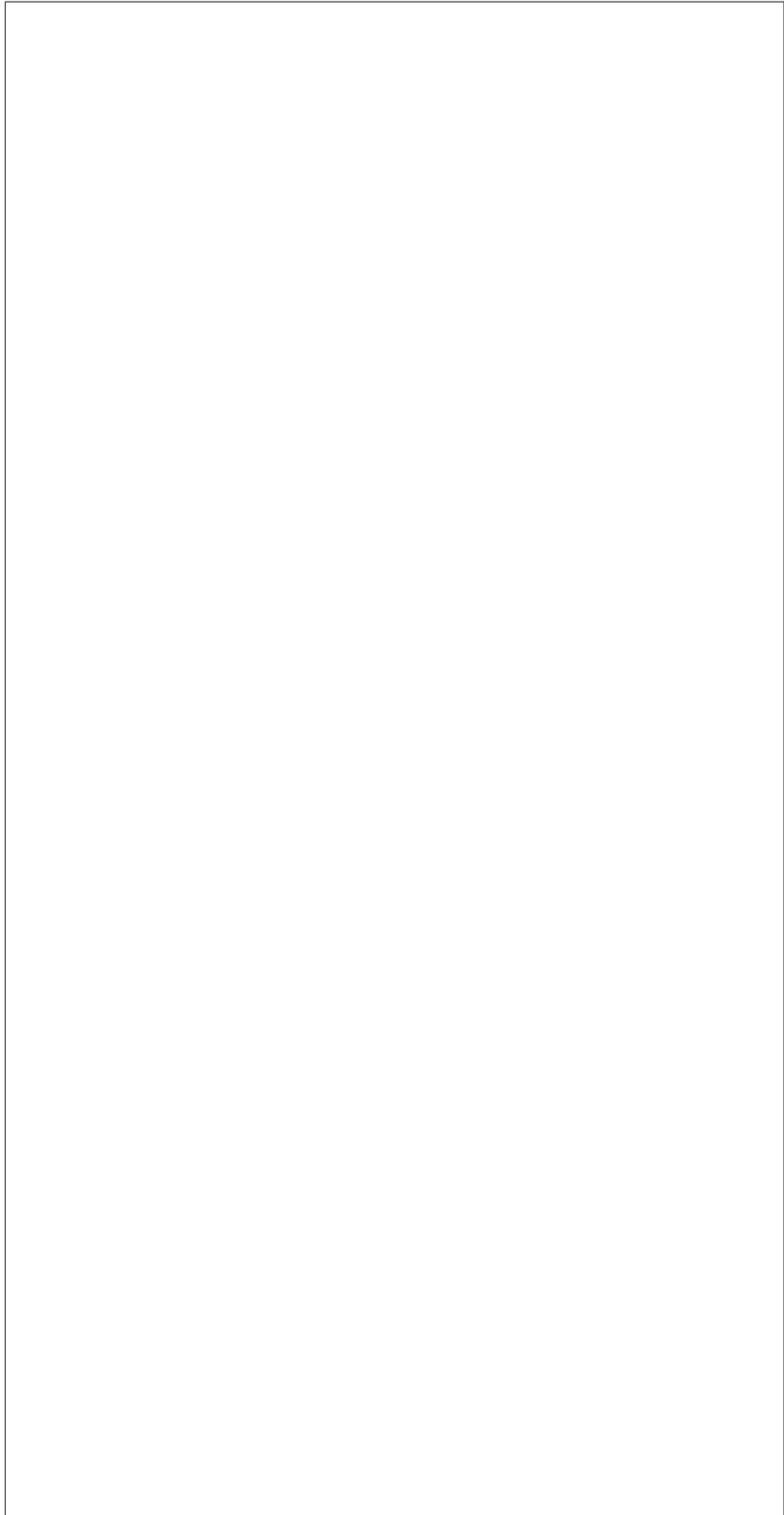
Re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic
can

be processed by Ceilometer. eg,



(continues on next page)

(continued from previous page)



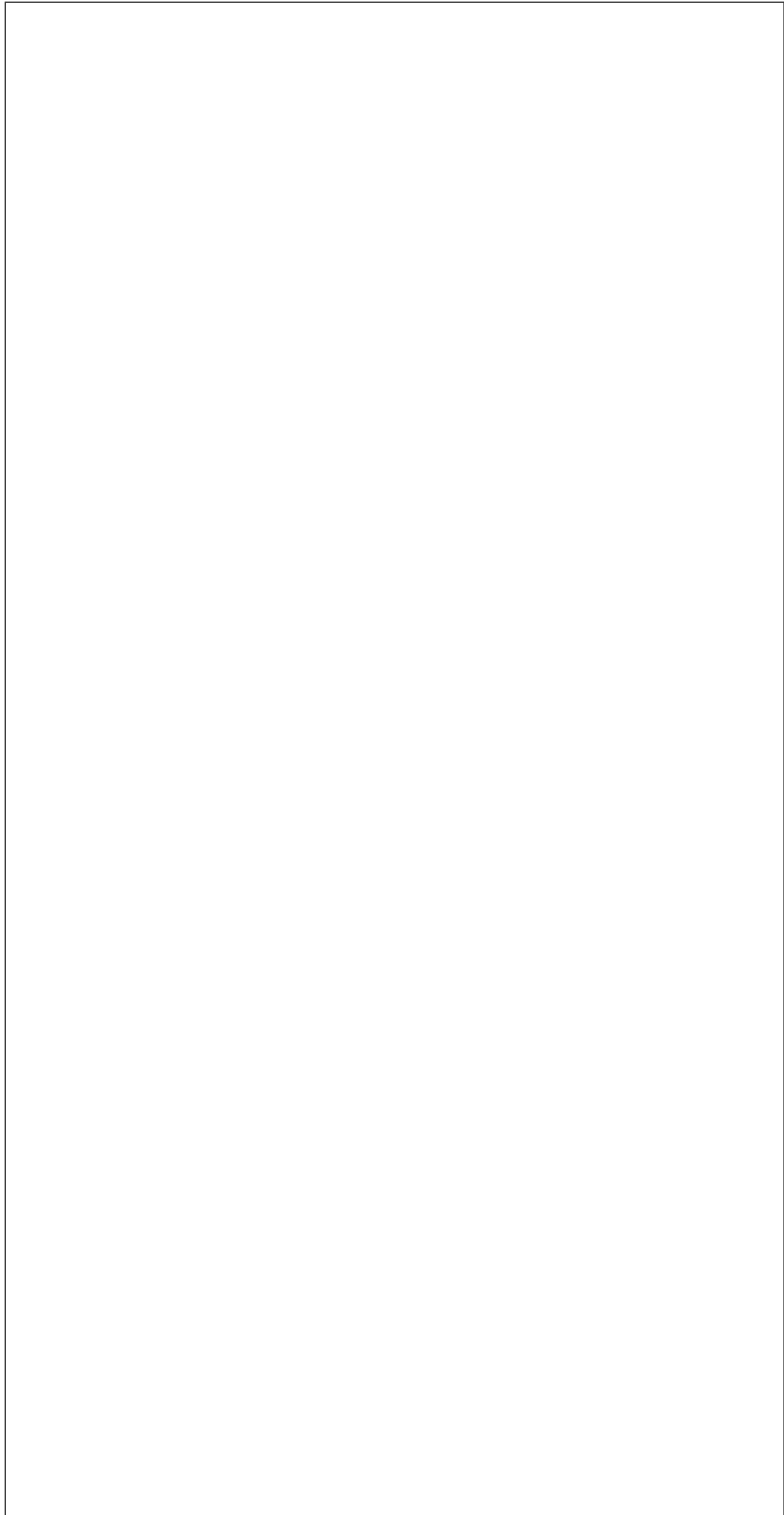
(continues on next page)

(continued from previous page)



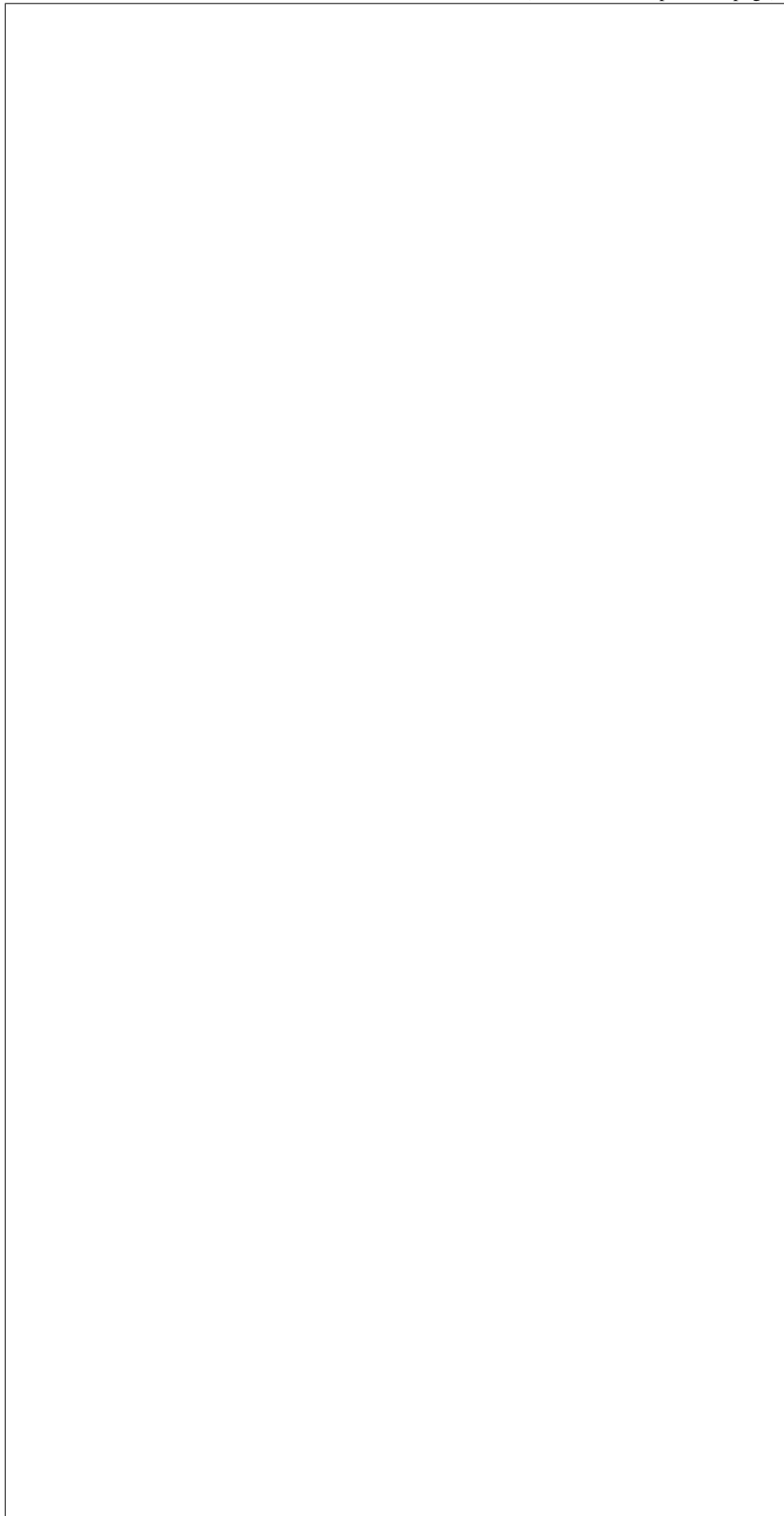
(continues on next page)

(continued from previous page)



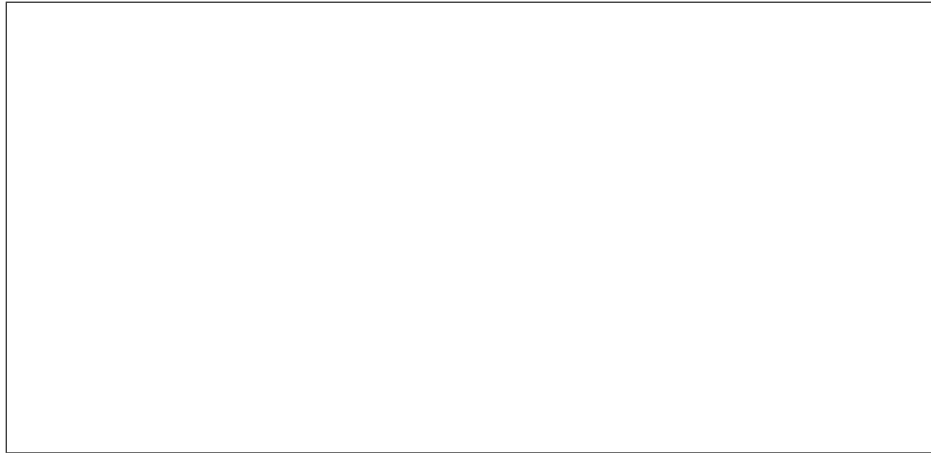
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



abstract

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter
task

A
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
[ironic](#)
[command](#)
[book](#)

get_suppr

Get
a
list
of
the
sup-
port
boot
mod

NOTE: I
may
not
im-
ple-
men
that.

Parameter
task
A
task
from
Task
ager

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
re-
ques
op-
er-
a-
tion
is
not
sup-
port
by

the driver

Raises
Driv
Op-
er-

a-
tion
or
its
deri-
tive
in
case
of
driv-
run-
time
er-
ror.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-
e-
ter
is
miss-
ing

Returns

A
list
with
the
sup-
port
boot
mod-
de-
fined
in
iro-
comm
boo-
If
boot

cannot be determined, empty list is returned.

mod
sup-
port

get_supp

Get
a
map
of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Parame

- **task**
A
task
from
Task
ager
- **comp**
If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-
tion
for
just
this
com
po-

nent, otherwise return indicators for all existing components.

Returns

A
dic-
tio-
nary
of
hard
ware
com-
po-
nent
(*irc*
com
com
as
keys
with
val-
ues
be-
ing

dictionaries having indicator IDs as keys and indicator properties as values.



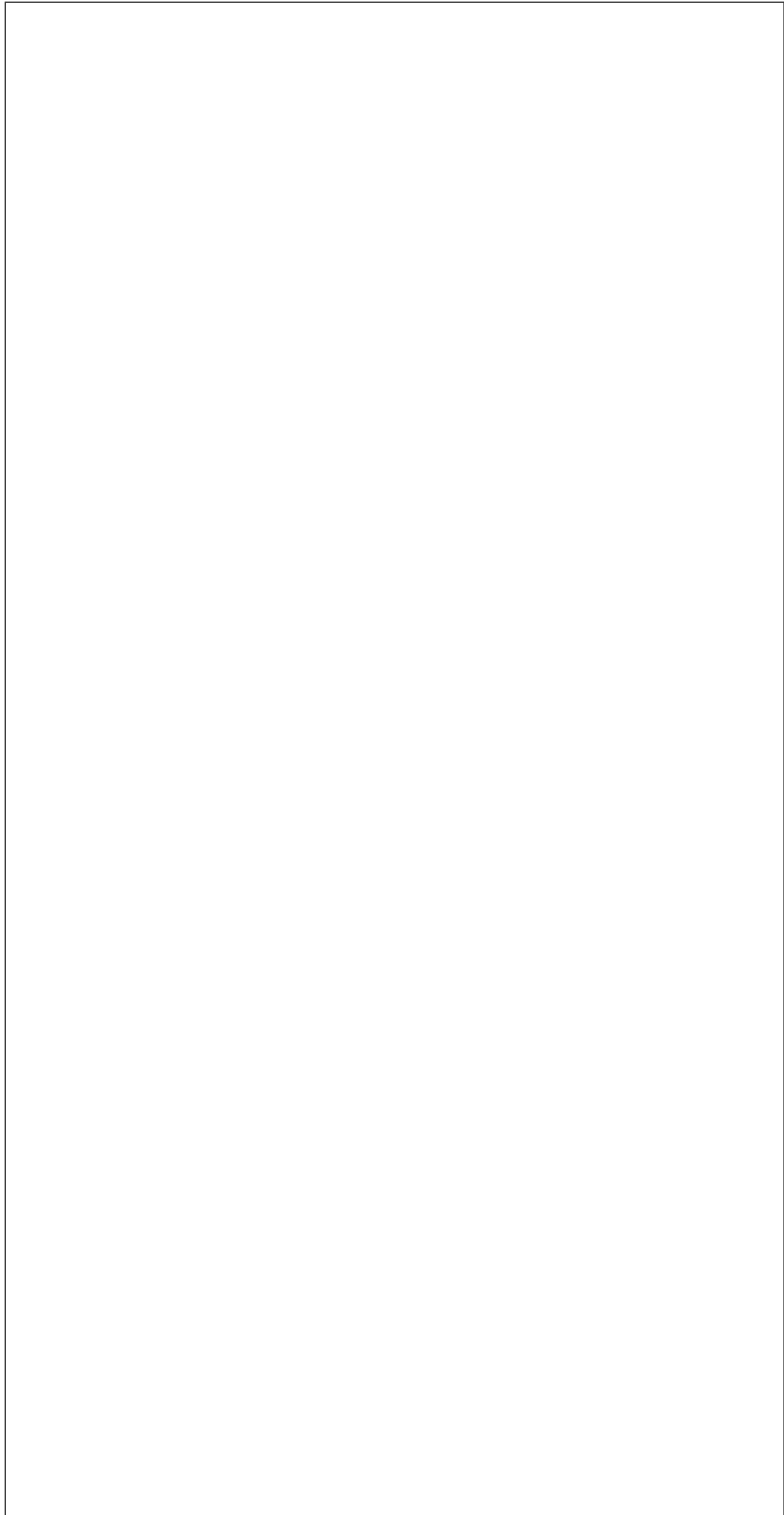
(continues on next page)

(continued from previous page)



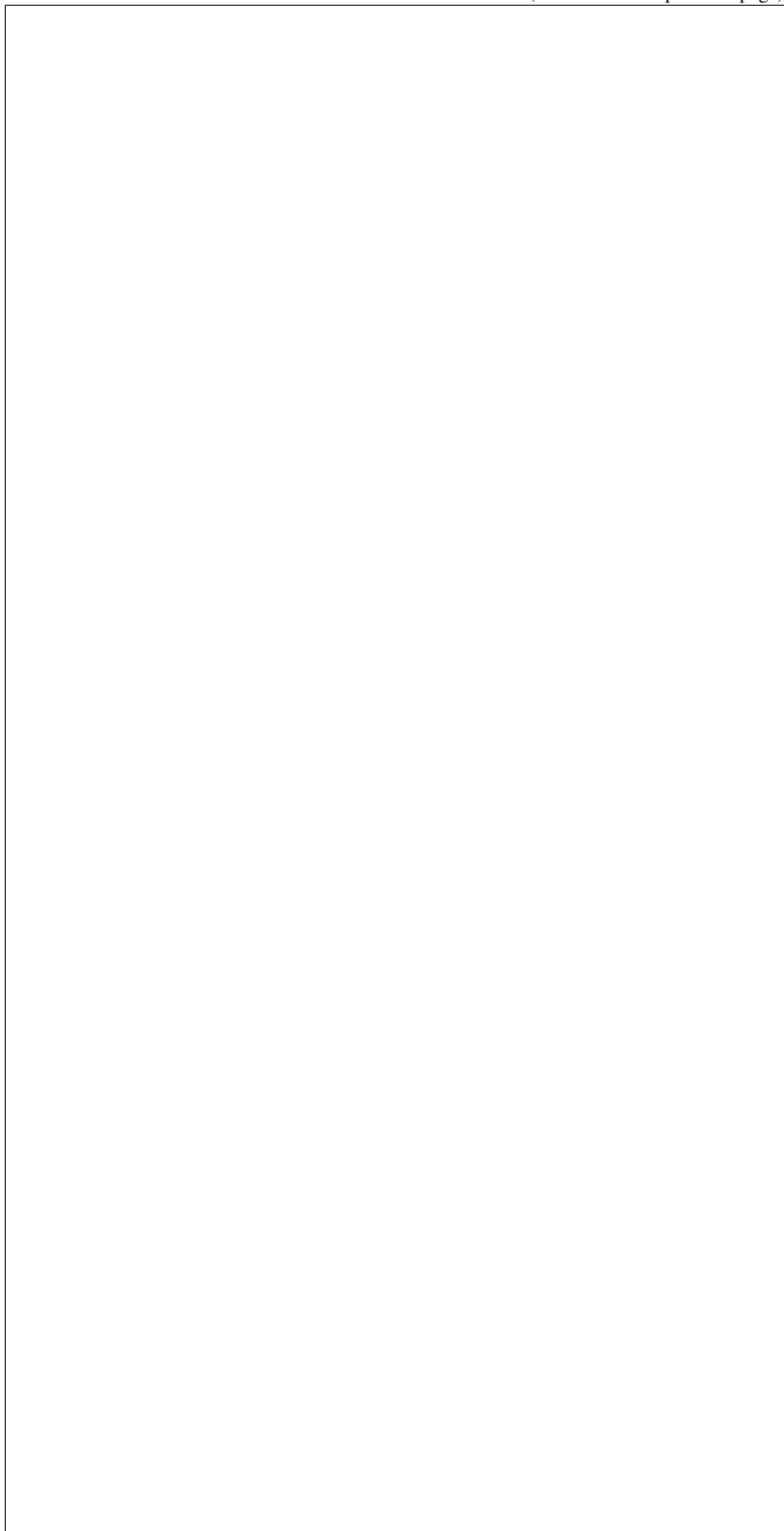
(continues on next page)

(continued from previous page)



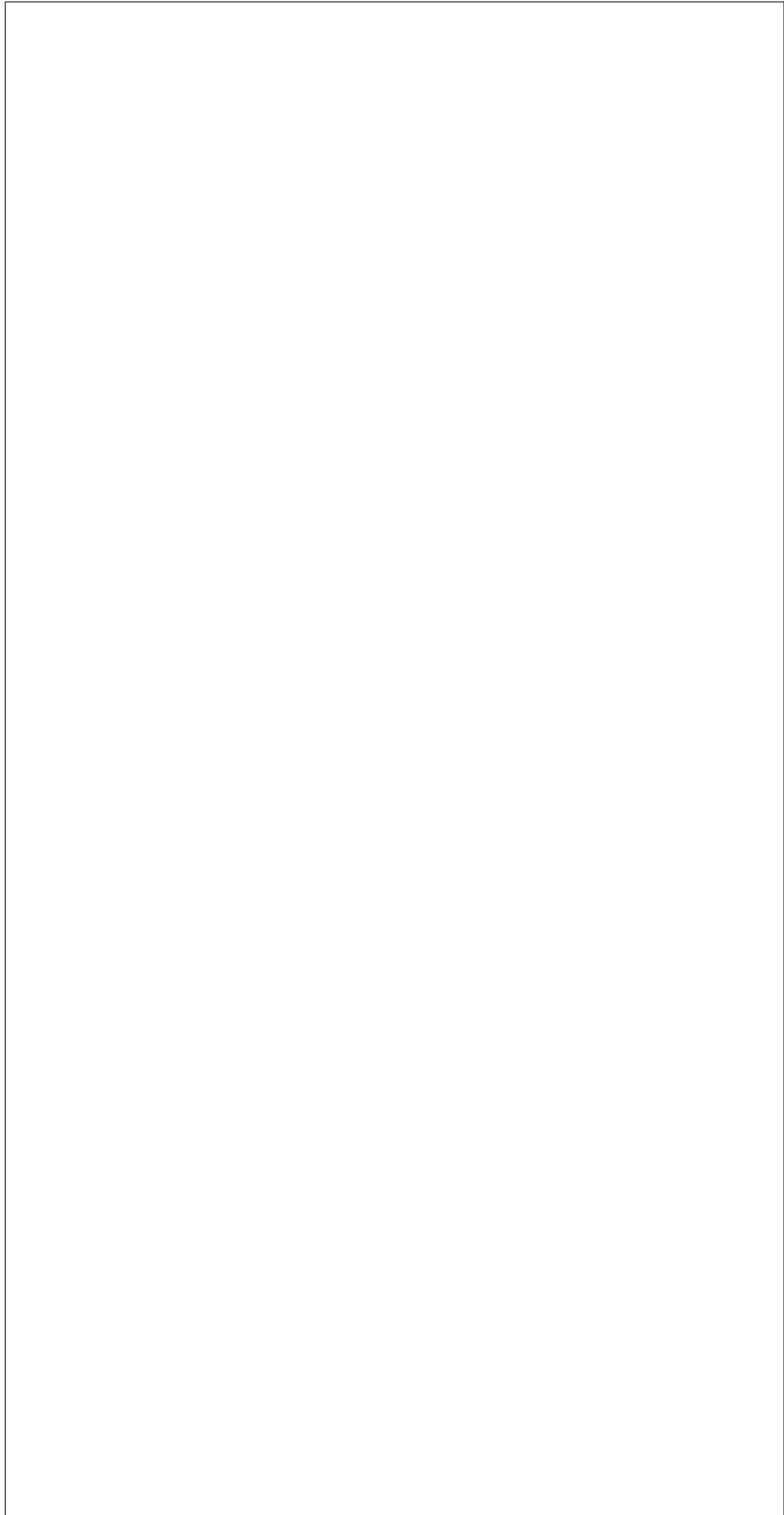
(continues on next page)

(continued from previous page)



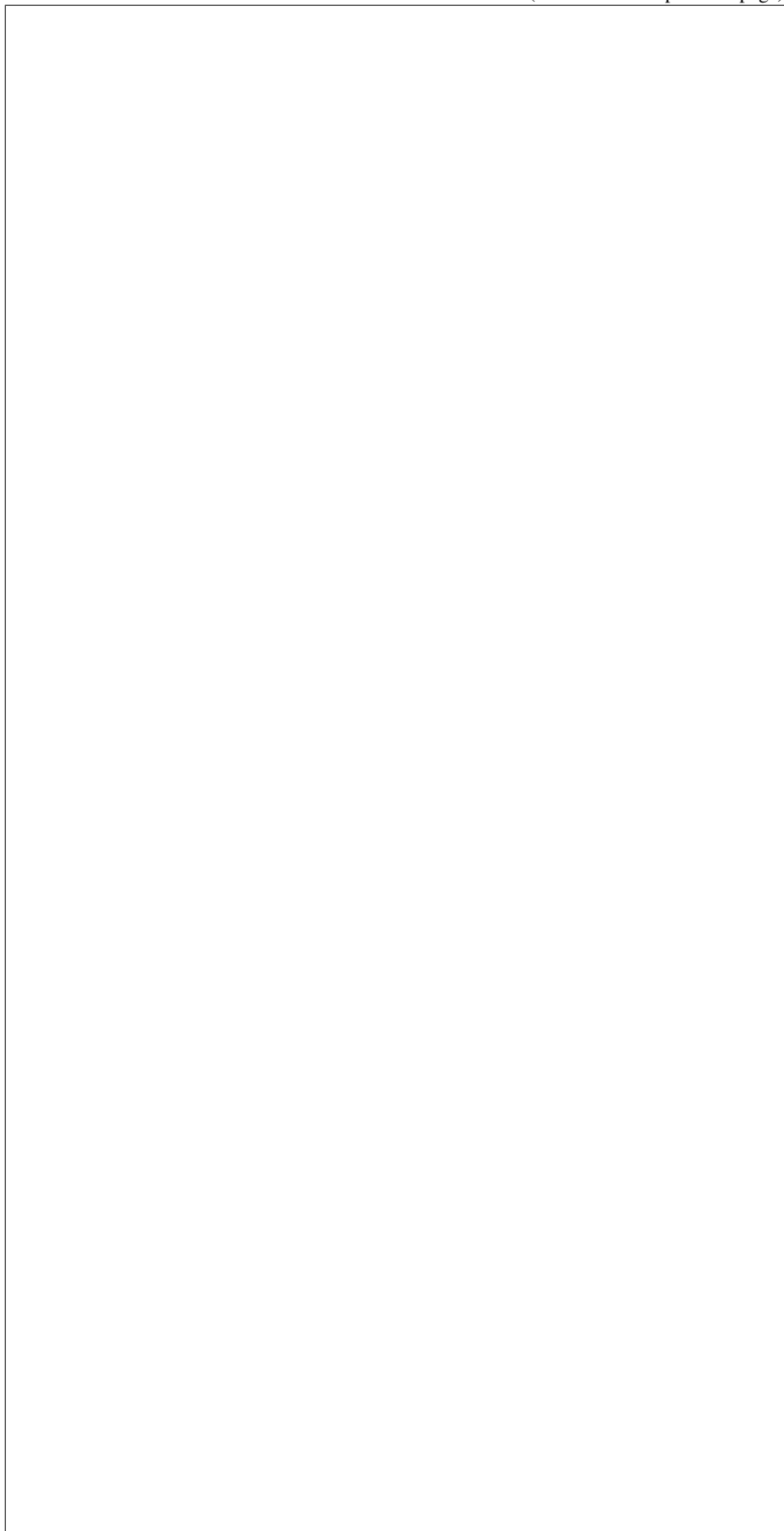
(continues on next page)

(continued from previous page)



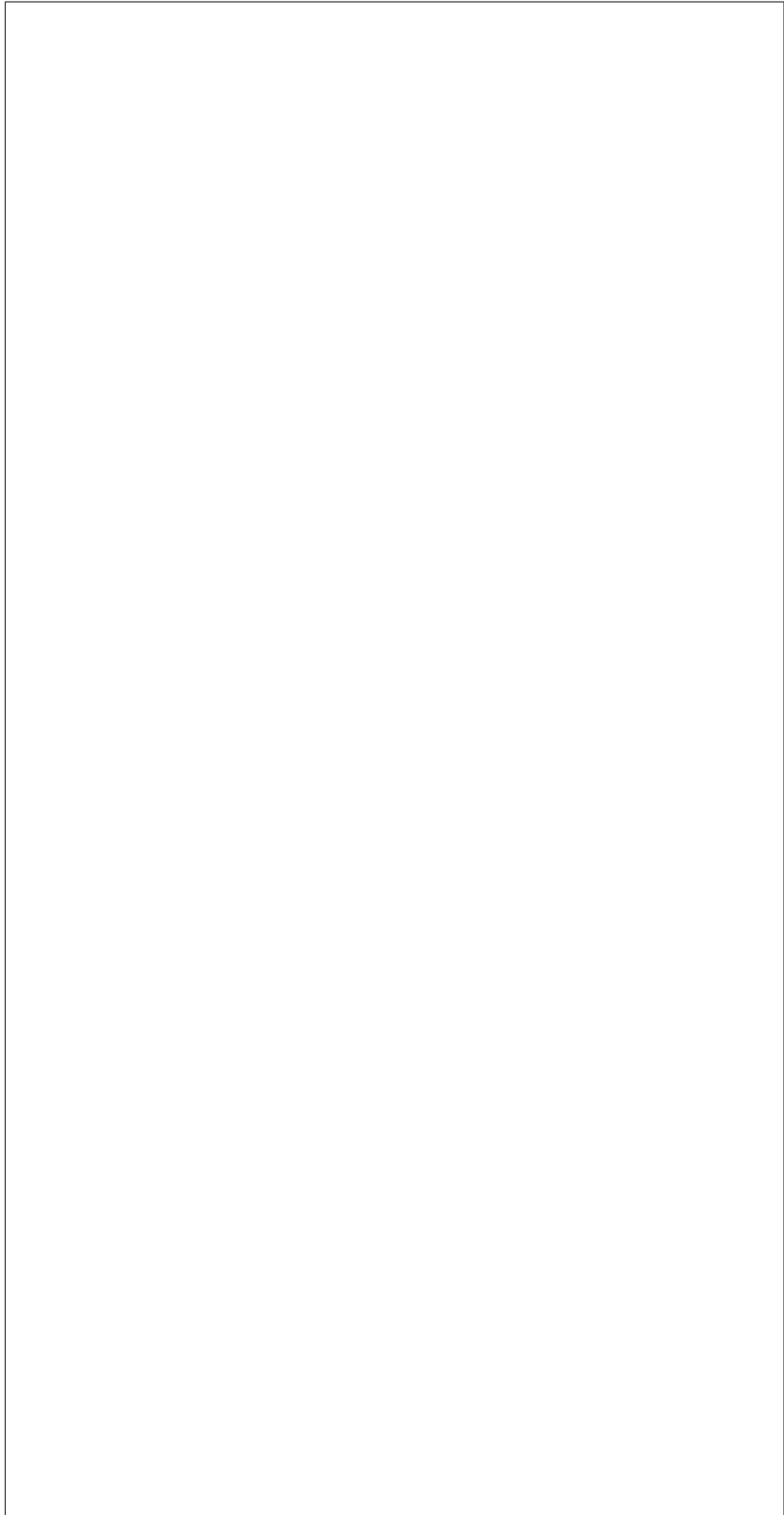
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_1

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Paramet
task
A

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

interface

In-
ter-
face
type
used
for
clear
steps
and
log-
ging

abstract

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use

on
next
re-
boot
of
the
node

Parame

- **task:**
A task from Task Manager.
- **dev:**
The boot device one of [ironic](#), [comm](#), [boo](#)
- **per:**
Boolean value True if the boot device will persist to all future boots. False if

not. Default: False.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod

to
use
on
next
re-
boot
of
the
node
Drive
im-
ple-
men
ing
this
meth
are
re-
quir
to
im-
ple-
men
the
get_
meth
as
well

NOTE: I
one
boot
mod
may
not
im-
ple-
men
that.

Parame

- **task**
A
task
from
Task
ager
-

mod
The
boot
mod
one
of
iro
comm
boo

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod
is
spec
i-
fied.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Un-
sup-
port

the driver

ed-
Driv
ten-
sion
if
re-
ques
op-
er-
a-
tion
is
not
sup-
port
by

Raises

Driv
Op-
er-
a-
tion
or
its
deri
tive
in
case
of
driv
run-
time
er-
ror.

set_ind:

Set
in-
di-
ca-
tor
on
the
hard
ware
com
po-
nent
to

the
de-
sired
state

Parameter

- **task**
A
task
from
Task
ager

- **comp**
The
hard
ware
com
po-
nent
one
of
iron
com
com

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

State

De-
sired
state
of
the
in-
di-
ca-
tor,
one

state is specified.

of
iro
com
ind

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-
nent
in-
di-
ca-
tor
or

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

set_sect

Set
the
cur-
rent

se-
cure
boot
state
for
the
node

NOTE: I
may
not
im-
ple-
men-
that.

Parame

- **task**
A
task
from
Task
ager
- **sta**
A
new
state
as
a
bool

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is

miss
ing

Raises

Driv
Op-
er-
a-
tion
or
its
deri
tive
in
case
of
driv
run-
time
er-
ror.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not
sup-
port
by
the
driv
or

the hardware

class ir

Base
iro
dri
bas
Bas
Base

class
for
net-
worl
in-
ter-
face

abstract

Add
the
clear
ing
net-
worl
to
a
node

Parameter

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror

add_insp

Add
the
in-
spec
tion

net-
worl
to
the
node

Parame

tas

A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

is invalid.

abstract

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Paramete

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

add_res

Add
the
res-
cu-
ing
net-
worl
to
the
node

Paramete

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary

in
the
form
{por
neu-
tron

Raises
Net-
worl
Er-
ror

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

is invalid.

abstract

Con
fig-
ure
ten-
ant
net-
worl
for
a
node

Parame
task
A
Task
ager

in-
stan

Raises

Net-
worl
Er-
ror

abstract

Re-
turn
the
cur-
rentl
used
VIF
as-
so-
ci-
ated
with
port
or
port
grou

We
are
boot
ing
the
node
only
in
one
net-
worl
at
a
time
and
pres
ence
of
clea
ing_

means were doing cleaning, of provisioning_vif_port_id - provisioning, of rescuing_vif_port_id - res-
cuing. Otherwise its a tenant network.

Parame

- **task**
A
Task
ager
in-
stan

- **p_obj**
Iron
port
or
port
grou
ob-
ject.

Returns

VIF
ID
as-
so-
ci-
ated
with
p_obj
or
Non

get_node

Re-
turn
net-
worl
con-
fig-
u-
ra-
tion
for
node
NIC

Gath
L2
and
L3
net-
worl
set-
ting

provider, then put together collected data in form of Nova network metadata (*network_data.json*) dict.

out-of-band.

from
iron
port
ob-
jects
and
un-
der-
ly-
ing
net-
work

Iron
wou
even
tu-
ally
pass
net-
work
con-
fig-
u-
ra-
tion
to
the
node
be-
ing
man
agec

Parame

task
A
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if

is invalid.

network metadata layout (*network_data.json*).

the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

Returns

a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
ad-
hear
ing
Nov

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

In-
ter-
face
type
used
for
clear
step
and
log-
ging

need_pow

Che
if
iron
node
mus
be
pow
ered
on
be-

fore
ap-
ply-
ing
net-
worl
char

Parame

tas

A
Task
ager
in-
stan

Returns

Boo

abstract

Han
dle
any
ac-
tions
re-
quir
whe
a
port
char

Parame

-

tas

A
Task
ager
in-
stan

-

por

a
char
Port
ob-
ject.

Raises

Con
flict.

Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstract

Han
dle
any
ac-
tion:
re-
quir
whe
a
port
chan

Parame

- **task**
A
Task
ager
in-
stan
- **por**
a
chan
Port
ob-
ject.

Raises

Con
flict.
Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstract

Re-

mov
the
clea
ing
net-
worl
from
a
node

Parame

tasl

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

remove_

Re-
mov
the
in-
spec
tion
net-
worl
from
a
node

Parame

tasl

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

Raises

In-

is invalid.

valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

abstract

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parame

task

A

Task

ager

in-

stan

remove_1

Re-

mov

the

res-

cu-

ing

net-

worl

from

a

node

Parame

task

A

Task

ager

in-

stan

Raises

Net-

worl

Er-

ror

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

the

net-

worl

in-

ter-

face

con-

fig-

is invalid.

u-
ra-
tion

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
som-
pa-
ram-
e-
ters
are
miss-
ing.

abstract

Un-
con-
fig-
ure
ten-
ant
net-
worl-
for
a
node

Parame

task
A
Task
ager
in-
stan-

validate

Val-
i-
date
the
net-
worl-
in-

is invalid.

ter-
face

Parame

task

A

Task

ager

in-

stan

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

the

net-

worl

in-

ter-

face

con-

fig-

u-

ra-

tion

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

if

som

pa-

ram-

e-

ters

are

miss

ing.

validate

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parame

task
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-

eters

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validate

Val-
i-
date
the
net-
worl
in-
ter-
face
for
res-
cue
op-
er-
a-
tion.

Parame

task
A
Task
ager
in-
stan

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-
face

is invalid.

con-
fig-
u-
ra-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

abstract

At-
tach
a
vir-
tual
net-
worl
in-
ter-
face
to
a
node

Parame

- **task**
A
Task
ager
in-
stan
- **vif**

whose value is a unique identifier for that VIF.

a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.
It
mus
have
an
id
key,

Raises

Net-
worl
Er-
ror,
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-
Port

abstract

De-
tach
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

- **task**
A
Task
ager
in-
stan

- **vif**
A
VIF
ID
to
de-
tach

Raises
Net-
worl
Er-
ror,
VifN
tAt-
tach

abstract

List
at-
tach
VIF
IDs
for
a
node

Parameter
task
A
Task
ager
in-
stan

Returns
List
of
VIF
dic-
tio-
nar-

ID of the VIF.

ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

class ir

Base
iron
dri
bas
Bas

In-
ter-
face
for
pow
relat
ac-
tion:

abstract

Re-
turn
the
pow
state
of
the
task
node

Paramete

task
A
Task
ager
in-
stan-
con-
tain-

ing
the
node
to
act
on.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

A
pow
state
One
of
iron
com
sta

get_supp

Get
a
list
of
the
sup-
port
pow
state

Parame

task
A
Task

ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
pow
state
de-
finec
in
iro
com
sta

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

abstract

Per-
form
a
hard
re-
boot
of
the
task
node

it on.

Driv
are
ex-
pect
to
prop
erly
han-
dle
case
whe
node
is
pow
ered
off
by
pow
er-
ing

Parame

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-

indicates to use default timeout.

ger
(>
0)
for
any
pow
state
Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

abstract

Set
the
pow
state
of
the
task
node

Parame

- **task**
A
Task
ager
in-
stan
con-
tain-

ing
the
node
to
act
on.

- **power**
Any
power
state
from
ironic
command
state

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
power
state
Non

indicates to use default timeout.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

e-
ter
is
miss
ing.

support:

Che
if
pow
sync
is
sup-
port
for
the
give
node

If
Fal
the
con-
duc-
tor
will
sim-
ply
store
wha
ever
get.
re-
turn
in
the
data
in-
stead

of trying to force the expected power state.

Parameter:

task
A
Task
ager
in-
stan-
con-
tain-
ing
the

node
to
act
on.

Returns

bool
when
power
sync
is
sup-
ported

class ir

Base
ironic
driver
base
Base

apply_co

Ap-
plies
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

fig-
u-
ra-
tion
to
ap-
ply.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

creating the new configuration.

del
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valic

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in

plete.

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

prog
asyn
chro
or
Non
if
it
is
com

abstract

Cre-
ates
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
It
as-
sum
that
the

Parame

ified in the nodes `target_raid_config`. Default value is `True`.

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

- **task**
A Task manager instance.
- **create_root_volume**
Setting this to `False` indicates not to create root volume that is specified.
- **create_non_root_volumes**
Setting this to `False` indicates not to create non-root volumes (all except

creating the new configuration.

chronously, or None if it is complete.

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn

abstract

Dele
RAI
con-
fig-
u-
ra-

ration is deleted, `node.raid_config` should be cleared by the implementation.

tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
Af-
ter
RAI
con-
fig-
u-

Parameters

task
A
Task
ager
in-
stan

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro

it is complete.

ified for logical disks and a textual description for them.

or
Non
if

get_log:

Get
the
prop
er-
ties
that
can
be
spec
i-
fied
for
log-
i-
cal
disk

This
meth
re-
turn
a
dic-
tio-
nary
con-
tain-
ing
the
prop
er-
ties
that
can
be
spec
i-

Returns

A
dic-
tio-
nary
con-
tain-

disks and a textual description for them.

ing
prop
er-
ties
that
can
be
men
tion
for
log-
i-
cal

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

interfa

In-
ter-
face
type
used
for
clea
step
and

log-
ging

validate

Val-
i-
date
the
RAI
In-
ter-
face

This
meth
val-
i-
date
the
prop
er-
ties
de-
fine
by
Iron
for
RAI
con-
fig-
u-
ra-
tion.

Driver implementations of this interface can override this method for doing more validations (such as BMCs credentials).

Parameter
task

A
Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters
are
miss
ing.

validate

Val-
i-
date
the
give
RAI
con-
fig-
u-
ra-
tion.
This
meth
val-
i-
date
the
give
RAI
con-

this interface can override this method to support custom parameters for RAID configuration.

fig-
u-
ra-
tion.
Driv
im-
ple-
men
ta-
tions
of

Parame

- **task**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-
ra-

menting an `apply_configuration` deploy step.

tion
is
in-
valid

ironic.d
{'create
indicate
'raid_co
'create_
to creat
'True'.
this to
creating
False},
'require
This
may
be
used
as
the
de-
ploy
args
ar-
gu-
men
for
RAI
in-
ter-
face
im-
ple-

class ir

Base
iro
dri
bas
Bas

In-
ter-
face
for
resc
relat
ac-
tion

out occurs.

clean_up

Clea
up
the
res-
cue
en-
vi-
ron-
men
for
the
task
node

This
is
par-
tic-
u-
larly
use-
ful
for
node
whe
res-
cu-
ing
is
asyn
chro
and
a
time

Paramete

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

Returns

Non

interfa

In-
ter-
face
type
used
for
clear
steps
and
log-
ging

abstract

Boo
the
task
node
into
a
res-
cue
en-
vi-
ron-
men

Parame

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

In-
stan-
cue-

Fail-
ure
if
node
val-
i-
da-
tion
or
res-
cue
op-
er-
a-
tion
fails

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it
is
com
plete

abstract

Tear
down
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to
nor-
mal.

ation fails.

Parameters

task

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Raises

In-

stan-

Un-

res-

cue-

Fail-

ure

if

node

val-

i-

da-

tion

or

un-

res-

cue

op-

er-

Returns

state

if

it

is

suc-

cess

ful.

class ironic

Base

ironic

driver

base

Bas

Base
class
for
stor-
age
in-
ter-
face

abstract

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Paramet

task
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

abstract

In-
form
the

stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parame

task

A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

abstract

De-
ter-
mine
if
de-
ploy

shou
per-
form
the
im-
age
writ
out.

Parame

tas

A
Task
ager
in-
stan

Returns

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect
the
im-
age
to
be
writ

ten by Ironic.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

class ir

Base
iro
dri

bas
Bas

In-
ter-
face
for
all
ven-
dor
pass
func
tion-
al-
ity.

Ad-
di-
tiona
veno

or
driv
spec
ca-
pa-
bil-
i-
ties
shou
be
im-
ple-
men
as
a

method in the class inheriting from this class and use the `@passthru` or `@driver_passthru` decorators.

Met
ods
ods
dec-
o-
ratec
with
`@dr`
shou
be
shor
livec
be-
caus
it
is

a
bloc
ing
call.

driver_v

Val-
i-
date
driv
vend
pass
ac-
tion

If
in-
valic
raise
an
ex-
cep-
tion
oth-
er-
wise
re-
turn
Non

Parame

- **metl**
meth
to
be
val-
i-
date
- **kwa**
info
for
ac-
tion.

Raises

Mis
ing-

Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain
cer-
tain
pa-
ram-
e-
ter.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
pa-
ram-
e-
ter
does
not
mate

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

abstract

Val-
i-
date

vend
spec
ac-
tion

If
in-
vali
rais
an
ex-
cep-
tion
oth-
er-
wise
re-
turn
Non

Parame

- **task**
A
task
from
Task
ager
- **met**
Met
to
be
val-
i-
date
- **kw**
Info
for
ac-
tion.

Raises

Un-
sup-
port
ed-
Driv
ten-

faces.

sion
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain
meth

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

class ir

Base
tup

metadata

Alia
for
field
num
ber
1

method

Alia
for
field
num
ber
0

ironic.d

A
dec-
o-
ra-
tor
to
cach
bios
set-
ting
af-
ter
run-
ning
the
func
tion.

Paramet

func
Func
tion
or
meth
to
wrap

ironic.d

Dec
o-
ra-
tor
for
clea
ing
step

Clea
ing
step
may
be

are ordered by priority from highest value to lowest value. For steps with the same priority, they are ordered by driver interface priority (see `conductor.steps.CLEANING_INTERFACE_PRIORITY`). `execute_clean_step()` will be called on each step.

used
in
man
ual
ual
or
au-
to-
mate
clea
ing.
For
au-
to-
mate
clea
ing,
only
step
with
pri-
or-
i-
ties
grea
than
0
are
used
The
step

For
man
ual
clea
ing,
the
clea
step
will
be
ex-
e-
cute
in
a
sim-
i-

to automated cleaning, but the steps and order of execution must be explicitly specified by the user when invoking the cleaning API.

ject. Clean steps used in manual cleaning may also take keyword variable arguments (as described in `argsinfo`).

turn *None* when finished, and the conductor will continue on to the next step. While the clean step is executing, the node will be in *states.CLEANING* provision state. If the step is asynchronous, the step should return *states.CLEANWAIT* to the conductor before it starts the asynchronous work. When the

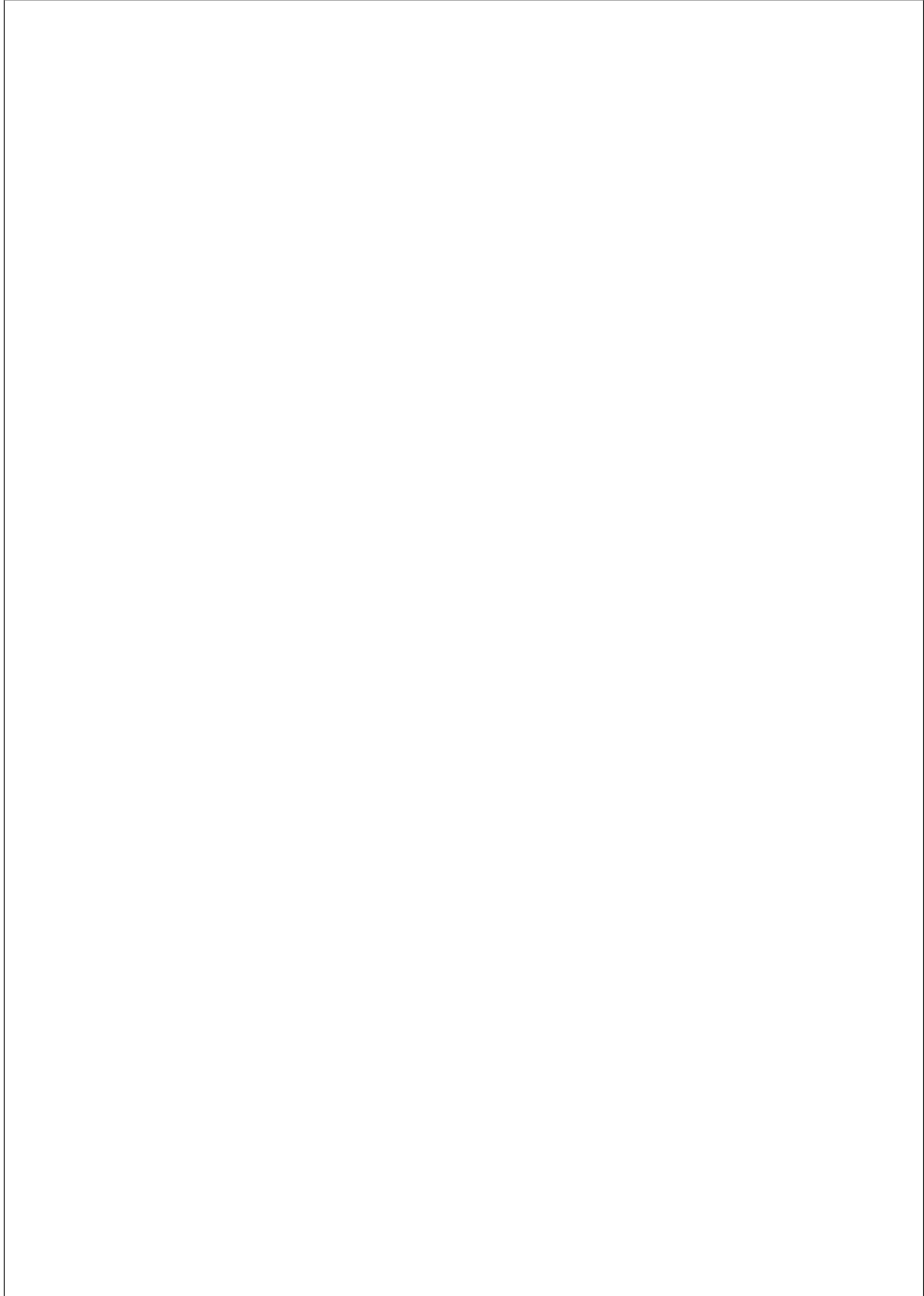
lar
fash
ion

Dec
o-
rate
clea
step
mus
take
as
the
only
po-
si-
tion
ar-
gu-
men
a
Task
ager
ob-

Clea
step
can
be
ei-
ther
syn-
chro
or
asyn
chro
If
the
step
is
syn-
chro
it
shou
re-

step is complete, the step should make an RPC call to *continue_node_clean* to move to the next step in cleaning. The node will be in *states.CLEANWAIT* provision state during the asynchronous work.

Ex-
am-
ples



(continues on next page)

(continued from previous page)

```
↪ {'size': {'description': 'size of widget (MB)',
```

(continues on next page)

```
↪ 'required': True}})
```


(continued from previous page)



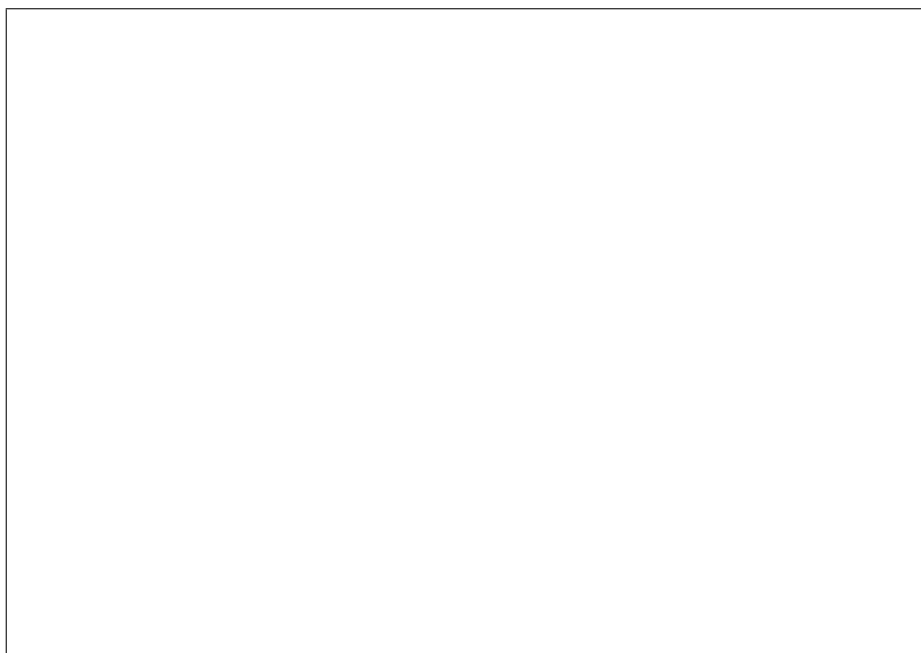
Parameter

- **priority**
an integer priority, should be a CON option
- **abort**
Boolean value. When

the
clea
step
is
abor
or
not;
de-
fault
to
Fals

- **arg**
a
dic-
tio-
nary
of
key-
wor
ar-
gu-
men
whe
key
is
the
nam
of
the
ar-

gument and value is a dictionary as follows:



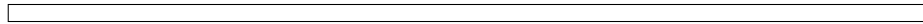
(continues on next page)

(continued from previous page)

↪it must be specified **in**

(continues on next page)

(continued from previous page)



out-of-band steps.

- **req**
When
this
step
re-
quir
the
rame
to
be
run-
ning
Sho
be
set
to
Fals
for
pure

Raises

Inv
if
any
of
the
ar-
gu-
men
are
in-
valid

ironic.d

Dec
o-
ra-
tor
for
de-
ploy
men
step
Only
step
with

ity from highest value to lowest value. For steps with the same priority, they are ordered by driver interface priority (see `conductor.steps.DEPLOYING_INTERFACE_PRIORITY`). `execute_deploy_step()` will be called on each step.

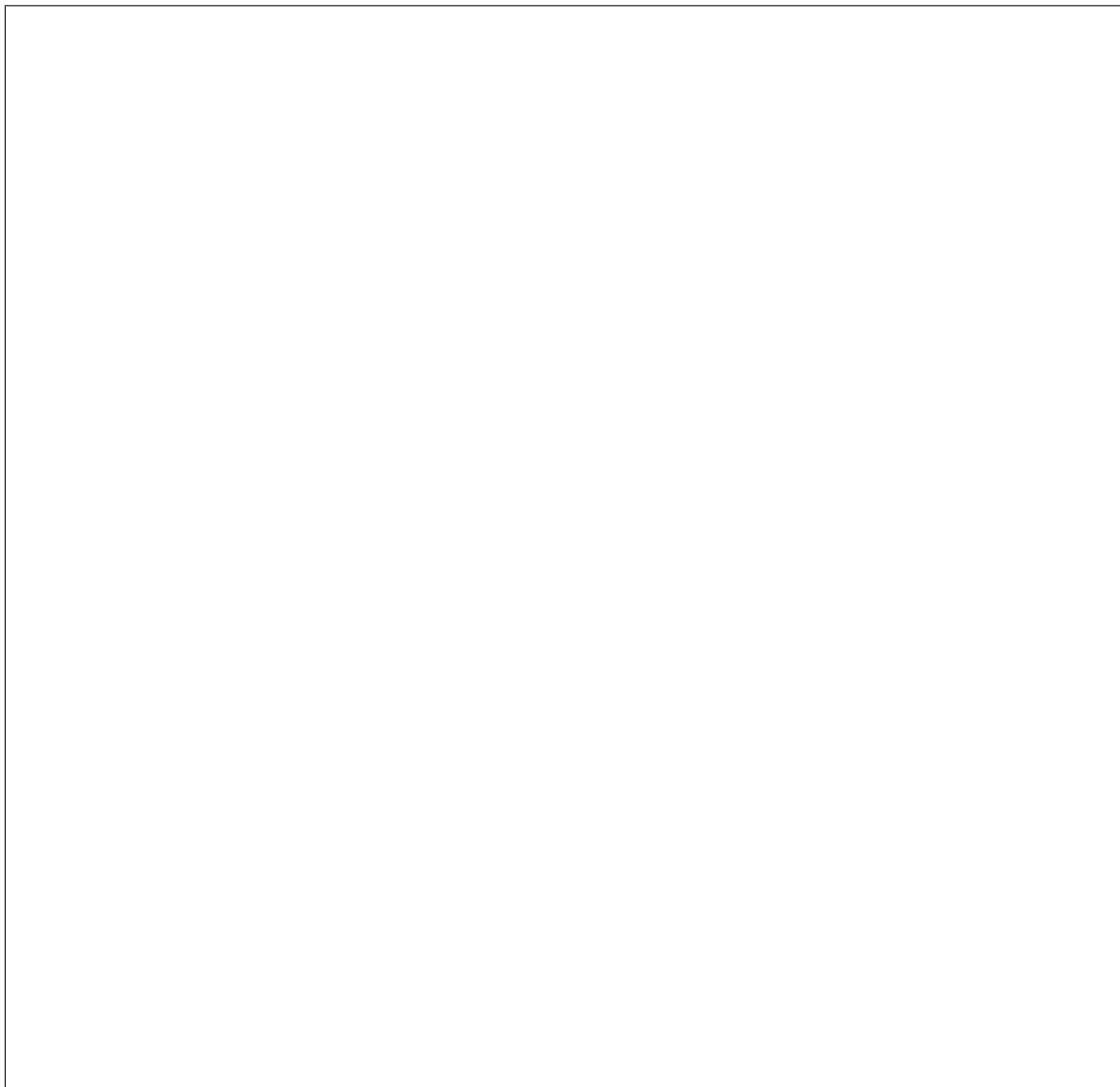
object.

pri-
or-
i-
ties
grea
than
0
are
used
The
step
are
or-
dere
by
pri-
or-

Dec
o-
rate
de-
ploy
step
mus
take
as
the
only
po-
si-
tion
ar-
gu-
men
a
Task
ager

De-
ploy
step
can
be
ei-
ther
syn-
chro

return *None* when finished, and the conductor will continue on to the next step. While the deploy step is executing, the node will be in *states.DEPLOYING* provision state. If the step is asynchronous, the step should return *states.DEPLOYWAIT* to the conductor before it starts the asynchronous work. When the step is complete, the step should make an RPC call to *continue_node_deploy* to move to the next step in deployment. The node will be in *states.DEPLOYWAIT* provision state during the asynchronous work.

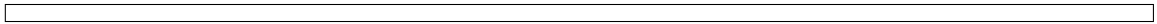


(continues on next page)

or
asyn
chro
If
the
step
is
syn-
chro
it
shou

Ex-
am-
ples

(continued from previous page)



which the step is run in the deployment process.

gument and value is a dictionary as follows:

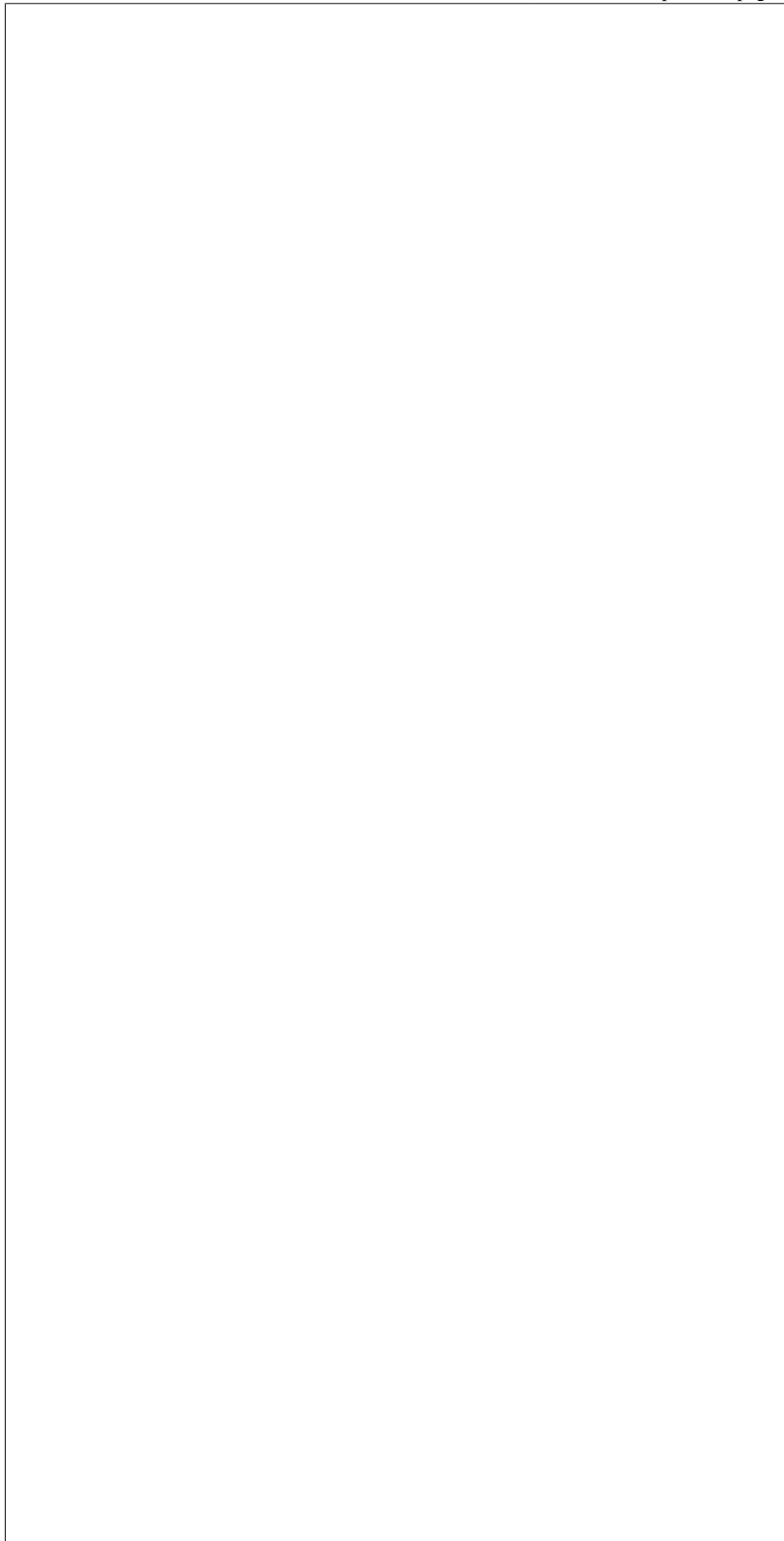


(continues on next page)

Paramet

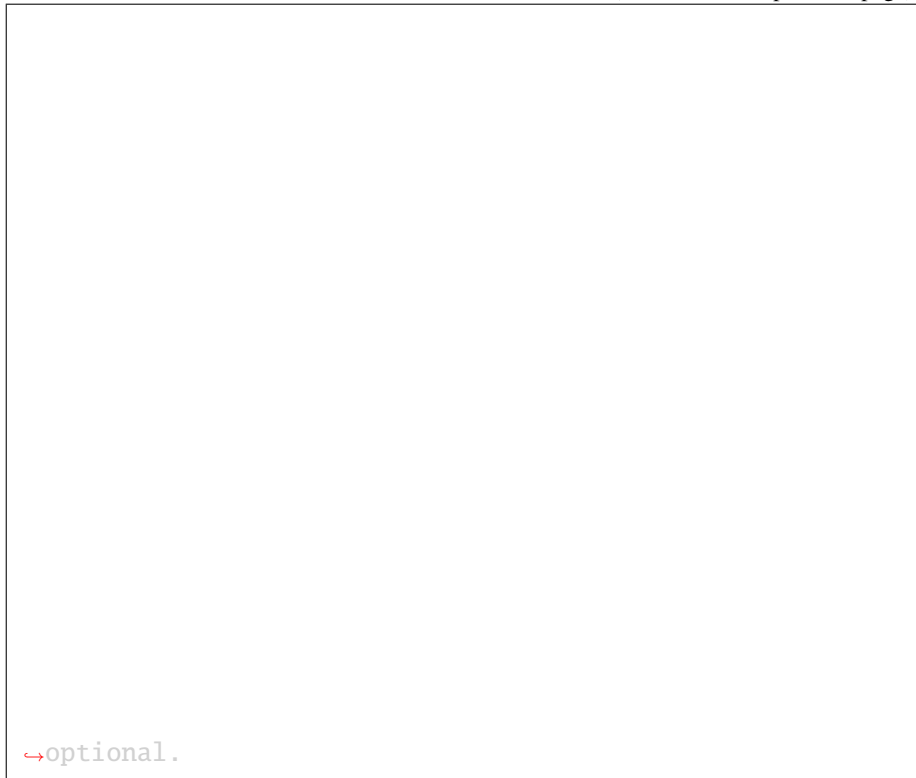
- **pri**
an
in-
te-
ger
(>=
pri-
or-
ity;
used
for
de-
ter-
min-
ing
the
or-
der
in
- **arg**
a
dic-
tio-
nary
of
key-
wor-
ar-
gu-
men-
whe
key
is
the
nam
of
the
ar-

(continued from previous page)



(continues on next page)

(continued from previous page)



Raises

Inv

if
any
of
the
ar-
gu-
men-
are
in-
valid

ironic.d

ironic.d

ironic.drivers.drac module

DRAC
Driv
for
re-
mote
sys-
tem
man
age-
men
us-
ing
Dell
Re-
mote
Ac-
cess
Card

class `ironic.drivers.drac.DellRemoteAccessCard`

Base
ironic.drivers.drac.DellRemoteAccessCard
Gen
in-
te-
grate
Dell
Re-
mote
Ac-
cess
Con
troll
hard
ware
type

property `ironic.drivers.drac.DellRemoteAccessCard.bios_interfaces`

List
of
sup-
port
bios
in-
ter-

face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port-
ven-
dor
in-
ter-
face

ironic.drivers.fake hardware module

Fake
hard
ware
type

class ir-

Base
ironic
drivers
hardware
Abstract

Fake
hard
ware
type

This
hard
ware
type
is
spec-
case
in
the
drive
fac-
tory
to
by-
pass
com-
pat-
i-
bil-
ity

verification. Thus, supported_* methods here are only for calculating the defaults, not for actual check.

All
fake
im-
ple-
men-
ta-
tions
are
still
ex-
pect
to
be
en-
able
in
the
con-
fig-
u-

ration.

property

List
of
class
of
sup-
port
bios
in-
ter-
face

property

List
of
class
of
sup-
port
boot
in-
ter-
face

property

List
of
class
of
sup-
port

con-
sole
in-
ter-
face

property

List
of
class
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
class
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
class
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
net-
work

in-
ter-
face

property

List
of
class
of
sup-
port
pow
in-
ter-
face

property

List
of
class
of
sup-
port
raid
in-
ter-
face

property

List
of
class
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
class
of
sup-
port
stor-
age
in-
ter-
face

property

List
of
class
of
sup-
port
res-
cue
in-
ter-
face

ironic.drivers.generic module

Gen
hard
ware
type

class ir

Base
ironic
drivers
hardware
Abstract

Ab-
strac
base
class
rep-
re-
sent
ing
gene
hard
ware

This
class
pro-
vide
rea-
son-
able
de-
fault
for
all
of

the
in-
ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
net-
work
in-
ter-
face

property

List
of
sup-
port

raid
in-
ter-
face

property

List
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
sup-
port
stor-
age
in-
ter-
face

class ir

Base
iron
drive
generator
Generator

Hardware
ware
type
that
uses
man-
ual
power
and
boot
man-
age-
ment

Us-
ing
this
hard-
ware
type

ting boot devices manually. This hardware type should only be used when no suitable hardware type exists in ironic, or the existing hardware type misbehaves for any reason.

as-
sum
that
an
op-
er-
a-
tor
man
ages
re-
boot
and
set-

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.hardware_type module

Ab-
strac
base
class
for
all
hard
ware
type

class ir
Base
obj

Ab-
strac
base
class
for
all
hard
ware
type

Hard
ware
type
is
a
fam-
ily
of
hard
ware
sup-
port
ing
the
sam
set
of
in-
ter-
face

from the ironic standpoint. This can be as wide as all hardware supporting the IPMI protocol or as narrow as several hardware models supporting some specific interfaces.

A
hard
ware

driver interface (power, deploy, etc).

type
de-
fines
an
or-
dere
list
of
sup-
port
im-
ple-
men
ta-
tions
for
each

get_prop

Get
the
prop
er-
ties
of
the
hard
ware
type

Note
that
this
re-
turn
prop
er-
ties
for
the
de-
fault
in-
ter-
face
of
each
type
for
this

hardware type. Since this is not node-aware, interface overrides cant be detected.

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

support

Whe
hard
ware
is
sup-
port
by
the
com
mu-
nity.

property

List
of
sup-
port
bios
in-
ter-
face

abstract

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port

con-
sole
in-
ter-
face

abstract

List
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

abstract

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
net-
worl
in-
ter-
face

abstract

List
of

sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
sup-
port
stor-
age
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.ibm module

iBM
Driv
for
man
ag-
ing
HUA
V5
se-
ries
rack
serv
such
as
228
V5,
CHI
V5.

class ir

Base
iro
dri
gen
Gen

Hua
iBM
hard
ware
type

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port

pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.ilo module

iLO
Driv
for
man
ag-
ing
HP
Pro-
liant
Gen
and
abov
serv

class ir

Base
ironic
drivers
ilo
Ilo

iLO
hard

ware
type
iLO
hard
ware
type
is
tar-
gete
for
iLO
base
Pro-
liant
Gen
serv

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

class `ironic`

Base
ironic

dri
gen
Gen

iLO
hard
ware
type

iLO
hard
ware
type
is
tar-
gete
for
iLO
4
base
Pro-
liant
Gen
and
Gen
serv

property

List
of
sup-
port
bios
in-
ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
con-

sole
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

ironic.drivers.intel_ipmi module

class ir

Base
iro
dri
ipm
IPM

In-
tel
IPM
hard
ware
type

Uses
ipm
to
im-
ple-
men
pow
and
man
age-
men
Pro-
vide
se-
rial
con-
sole
im-
ple-
men

tations via shellinabox or socat. Supports Intel SST-PP feature.

property

List
of
sup-
port
man
age-
men
in-
ter-
face

ironic.drivers.ipmi module

Hardware
ware
type
for
IPM
(us-
ing
ip-
mi-
tool)

class `ironic.drivers.ipmi`

Base
ironic
drivers
generic
Gen

IPM
hard
ware
type

Uses
ipm
to
im-
ple-
men
pow
and
man
age-
men

Pro-
vide
se-
rial
con-
sole
im-
ple-
men

tations via shellinabox or socat.

property

List
of
sup-
port

con-
sole
in-
ter-
face

property

List
of
sup-
port
man-
age-
men-
in-
ter-
face

property

List
of
sup-
port
pow-
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.irmc module

iRM
Driv
for
man-
ag-
ing
FU-
JITS
PRI
BX
S4

JITSU PRIMERGY servers, and above servers.

tem.

or
RX
S8
gen-
er-
a-
tion
of
FU-

class `ironic`
Base
ironic
dri
gen
Gen

iRM
hard
ware
type

iRM
hard
ware
type

is
tar-
gete
for
FU-

JITS
PRI
serv

whic
have
iRM
S4

man
age-
men
sys-

property

List
of
sup-
port
bios
in-
ter-

face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
con-
sole
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man-
age-
men-
in-
ter-
face

property

List
of
sup-
port
pow-
in-
ter-

face

property

List
of
sup-
port
raid
in-
ter-
face

ironic.drivers.redfish module

class ir

Base
ironic
driver
generator
Gen

Red-
fish
hard-
ware
type

property

List
of
sup-
port
bios
in-
ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port

pow
in-
ter-
face

property

List
of
sup-
port
man-
age-
men-
in-
ter-
face

property

List
of
sup-
port
pow-
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.snmp module

SNM
hard
ware
type

class ir

Base
ironic
driver
generator
Gen

SNM
Hard
ware
type

property

List
of
sup-
port
man-
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

ironic.drivers.utils module

class ir

Base
ironic
driver
base
Ven

Wra
per
arou
mul-
ti-
ple
Ven-
dor-
In-
ter-
face

get_prop

Re-
turn
the
prop
er-
ties
from
all
the
Ven-
dor-
In-
ter-
face

Returns

a
dic-
tio-
nary
of
<pro
erty.
en-
tries

validate

Call
val-
i-
date
on
the
ap-
pro-
pri-
ate

faces.

in-
ter-
face
only

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
meth
is
in-
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
meth

or
pa-
ram-
e-
ters
in
kwa

ironic.d

Add
ca-
pa-
bil-
ity
to
node
ca-
pa-
bil-
i-
ties
prop
erty.

If
ca-
pa-
bil-
ity
is
al-
read
pres
then
a
du-
pli-
cate
en-
try
will
be
add

Paramet

- **task**
Task
ob-
ject.

- **cap**
Ca-
pa-
bil-
ity
key.
- **valu**
Ca-
pa-
bil-
ity
valu

ironic.d

Pars
the
ca-
pa-
bil-
i-
ties
strin
into
a
dic-
tio-
nary

Paramet

cap
the
ca-
pa-
bil-
i-
ties
of
the
node
as
a
for-
mat-
ted
strin

Raises

In-
valid

a malformed value

Pa-
ram-
e-
ter-
Valu
if
ca-
pa-
bil-
i-
ties
is
not
an
strin
or
has

ironic.d

Col-
lect
and
store
the
sys-
tem
logs
from
the
IPA
rame

Col-
lect
and
store
the
sys-
tem
logs
from
the
IPA
rame
This
meth
mak
a
call
to

ramdisk to collect the logs and store it according to the configured storage backend.

the
IPA

Parameter

- **node**
A node object.
- **label**
A string to label the log file such as a clear step name

ironic.d

Ensure boot from correct device if persistent is True
If ipmi is True and is_n

set
to
boot
from
cor-
rect
de-
vice
else
un-
set
is_n
field

Paramet

- **task**
Node
ob-
ject.
- **drive**
Node
drive

`ironic.d`

Set
per-
sis-
tent
boot
de-
vice
to
drive
If
per-
sis-
tent
is
True
set
per-
sis-
tent
field
to
the
boot

sistent to False, else set `is_next_boot_persistent` to False.

de-
vice
and
re-
set
per-

Parameter

- **task**
Task
ob-
ject.
- **device**
Boo
de-
vice
- **persistent**
Whe
next
boot
is
per-
sis-
tent
or
not.

`ironic.d`

Get
the
agen
ISO
im-
age.

`ironic.d`

Get
the
agen
ker-
nel/
as
a
dic-

tio-
nary
ironic.d

Get
the
ap-
pli-
ca-
ble
ker-
nel
para
The
lo-
ca-
tions
are
check
in
this
or-
der:

1. The node instance
2. The node driver
3. Con-
fig-
u-
ra-
tion.

Paramet

- **node**
Node
ob-
ject.
- **def**

De-
fault
valu

ironic.d

Re-
turn
ca-
pa-
bil-
ity
valu
from
node
ca-
pa-
bil-
i-
ties
prop
erty.

Paramet

- **node**
Nod
ob-
ject.
- **cap**
Ca-
pa-
bil-
ity
key.

Returns

Ca-
pa-
bil-
ity
valu
If
ca-
pa-
bil-
ity
is
not

pres
then
re-
turn
Non

ironic.d

Get
all
MA
ad-
dres
for
the
port
be-
long
ing
to
this
task
node

Paramet

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
of
MA
ad-
dres
in
the
for-
mat
xx:x

ironic.d

Con
struc
the
log
file
nam

Paramet

- **node**
A
node
ob-
ject.

- **label**
A
strin
to
la-
bel
the
log
file
such
as
a
clear
step
nam

Returns

The
log
file
nam

ironic.d

Re-
mov
-

and
:
char
ac-
ters
and

low-
er-
case
the
MA
strin

Paramet

mac
MA
ad-
dres
to
nor-
mal-
ize.

Returns

Nor-
mal-
ized
MA
ad-
dres
strin

ironic.d

Stor
the
ram
logs

This
meth
store
the
ram
logs
ac-
cord
ing
to
the
con-
fig-
ured
stor-
age
back
end.

Paramet

- **node**
A
node
ob-
ject.

- **log**
A
gzip
and
base
en-
code
strin
con-
tain-
ing
the
logs
arch

- **label**
A
strin
to
la-
bel
the
log
file
such
as
a
clea
step
nam

Raises
OS-
Er-
ror
if
the
di-
rec-
tory
to
save
the
logs

can-
not
be
cre-
ated

Raises

IO-
Er-
ror
when
the
logs
cannot
be
saved
to
the
local
file
system.

Raises

Swi-
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi-
fails

ironic.drivers.xclarity module

XCl-
ity
Drive
and
sup-
port
ing
meta-
class

class `ironic.objects.drivers.XCL`
Base
ironic
drivers
generation
Generation
XCL
ity
hard
ware
type

property
List
of
sup-
port
man-
age-
men-
in-
ter-
face

property
List
of
sup-
port
pow-
in-
ter-
face

Module contents

`ironic.objects` package

Submodules

`ironic.objects.allocation` module

class `ironic.objects.base`
Base
ironic
objects
base

Iro
osl
bas
Ver

VERSION

property

property

create(C

Cre-
ate
a
Al-
lo-
ca-
tion
reco
in
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Raises

Al-

lo-
ca-
tion
pli-
cate
Nam
Al-
lo-
ca-
tion
Al-
read
ists

property

dbapi =

destroy

Dele
the
Al-
lo-
ca-
tion
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A

context should be set when instantiating the object, e.g.: Allocation(context)

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

fields =

'oslo_ve
'conduct
'oslo_ve
'create
'oslo_ve
'extra'
'oslo_ve
Integer
'oslo_ve
'last_er
'oslo_ve
String(
'oslo_ve
'node_id
'oslo_ve
'owner'
'oslo_ve
'resourc
'oslo_ve
'state'
'oslo_ve
'traits
'oslo_ve
'update
'oslo_ve
UUID(de
'oslo_ve

classmet

Find
an
al-
lo-
ca-
tion
by
its
ID,

UUI
or
nam

Parame

- **all**
The
ID,
UUI
or
nam
of
an
al-
lo-
ca-
tion.

- **con**
Se-
cu-
rity
con-
text

Returns

An
ALL
ob-
ject.

Raises

In-
va-
li-
dI-
den-
tity

classme

Find
an
al-
lo-
ca-
tion
by
its
in-

te-
ger
ID.

Parame

- **cls**
the
ALL

- **con**
Se-
cu-
rity
con-
text

- **all**
The
ID
of
an
al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Al-
lo-
ca-
tion-
Not-
Four

classmet

Find
an
al-
lo-
ca-
tion
base
by

its
nam

Parame

- **cls**
the
ALL
- **con**
Se-
cu-
rity
con-
text
- **nam**
The
nam
of
an
al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Al-
lo-
ca-
tion-
Not-
Four

classmet

Find
an
al-
lo-
ca-
tion
by
its
UUI

Parameters

- **cls**
the
All
- **con**
Se-
cu-
rity
con-
text
- **uui**
The
UUI
of
an
al-
lo-
ca-
tion.

Returns

An
All
ob-
ject.

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

property

classmethod

Re-
turn
a
list
of
Al-

lo-
ca-
tion
ob-
jects

Parame

- **cls**
the
All

- **con**
Se-
cu-
rity
con-
text.

- **fil**
Fil-
ters
to
ap-
ply.

- **lim**
Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
Pag-
i-

na-
tion
marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

Returns

A
list
of
ALL
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

property

property

property

refresh

Load
up-
date
for
this
Al-
lo-
ca-
tion.

Load
an
al-
lo-
ca-
tion
with
the
same
uuid
from
the
data
and
check
for
up-
date
at-
tribu

Updates are applied from the loaded allocation column by column, if there are any updates.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

in-
di-

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

save(*con*)

Save
up-
date
to
this
Al-
lo-
ca-
tion.

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Al-
lo-
ca-
tion-
Not-
Four
Al-
lo-
ca-
tion
pli-
cate
Nam

property

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion

when
iron
cre-
ates,
up-
date
or
dele
an
al-
lo-
ca-
tion.

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci:
'warning
'oslo_ve
'publish
'oslo_ve
'update
'oslo_ve

property

property

property

property

class ir

Base
iron
obj
not
Not

```
SCHEMA =  
'created'  
'extra')  
( 'allocat  
'resource'  
( 'allocat  
'updated'  
'uuid') )
```

```
VERSION
```

```
property
```

```
property
```

```
property
```

```
fields =
```

```
'oslo_ve  
'created'  
'oslo_ve  
'extra'  
'oslo_ve  
'last_en  
'oslo_ve  
String(  
'oslo_ve  
'node_u  
'oslo_ve  
'owner'  
'oslo_ve  
'resourc  
'oslo_ve  
'state'  
'oslo_ve  
'traits'  
'oslo_ve  
'updated'  
'oslo_ve  
UUID(de  
'oslo_ve
```

```
property
```

```
property
```

```
property
```

property

property

property

property

property

property

ironic.objects.base module

Iron
com
mon
in-
ter-
nal
ob-
ject
mod

class ir

Base
osl
bas
Ver

Base
class
and
ob-
ject
fac-
tory.

This
form
the
base
of
all

ob-
jects
that
can
be
re-
mote
or
in-
stan-
ti-
ated
via
RPC

Simply defining a class that inherits from this base class will make it remotely instantiatable. Objects should implement the necessary get classmethod routines as well as save object methods as appropriate.

OBJ_PROJ

OBJ_SER

as_dict

Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is
JSON
serial

convert

Con-
vert
this
ob-
ject

older, or newer than the version of the object. This is used for DB interactions as well as for serialization/deserialization.

to
the
tar-
get
ver-
sion
Con
vert
the
ob-
ject
to
the
tar-
get
ver-
sion
The
tar-
get
ver-
sion
may
be
the
sam

The
re-
mov
flag
is
used
to
dis-
tin-
guis
thes
two
case

1)
For
se-
ri-
al-
iza-
tion
we
need

service receiving the object may not know about these fields. `remove_unavailable_fields` is set to `True` in this case.

appropriate values so that these fields are saved in the DB. (If they are not set, the `VersionedObject` magic will not know to save/update them to the DB.) `remove_unavailable_fields` is set to `False` in this case.

to
re-
mov
the
un-
avai
able
field
be-
caus
the

2)
For
DB
in-
ter-
ac-
tion
we
need
to
set
the
un-
avai
able
field
to
their
ap-
pro-

_con
vert
does
the
ac-
tual
work

Parameter

- **target**
the
de-
sired

to True when (de)serializing. False to set the unavailable fields to appropriate values; set this to False for DB interactions.

ver-
sion
of
the
ob-
ject

•
rem
True
to
re-
mov
field
that
are
un-
avai
able
in
the
tar-
get
ver-
sion
set
this

do_vers:

Cha
the
ob-
ject
to
the
ver-
sion
need
for
the
data

If
need
this
char
the
ob-
ject
(mo

sion for saving to the database.

i-
fies
ob-
ject
field
to
be
in
the
cor-
rect
ver-

The
ver-
sion
used
to
save
the
ob-
ject
in
the
DB
is
de-
ter-
mine
as
fol-
lows

- If the object is pinned we save the object in the pinned version. Since

pinned, we must not save in a newer version, in case a rolling upgrade is happening and some services are still using the older version of ironic, with no knowledge of this newer version.

must only be called just before saving the object to the DB.

it
is

- If the object isn't pinned we save the object in the latest version. Because the object may be converted to a different object version this meth

Returns
a dictionary of character

These are the fields/values of the object that would be saved to the DB.

field
and
their
new
val-
ues
(cou
be
an
emp
dic-
tio-
nary

```
fields =  
'oslo_ve  
'updatee  
'oslo_ve
```

classmet

Re-
turn
the
tar-
get
ver-
sion
for
this
ob-
ject.

This
is
the
ver-
sion
in
whic
the
ob-
ject
shou
be
ma-
nip-
u-
latec
e.g.
sent
over

wire via RPC or saved in the DB.

erwise, returns the version of the object.

the

Returns

if
pinn
re-
turn
the
ver-
sion
of
this
ob-
ject
cor-
re-
spor
ing
to
the
pin.
Oth-

Raises

ovo_

obj_refr

Ap-
plies
up-
date
for
ob-
jects
that
in-
herit
from
base

Che
for
up-
date
at-
tribu
in
an
ob-
ject.

Up-
date
are
ap-
plied
from
the
load
ob-
ject

column by column in comparison with the current object.

classmet

Re-
turn
when
this
ob-
ject
sup-
port
a
par-
tic-
u-
lar
ver-
sion

Che
the
re-
ques
ver-
sion
agai
the
ob-
jects
tar-
get
ver-
sion
The
tar-
get
ver-
sion
may

not be the latest version during an upgrade, when object versions are pinned.

Parame

ver:
A
tu-
ple
rep-
re-
sent
ing
the
ver-
sion
to
chec

Returns
When
the
ver-
sion
is
sup-
port

Raises
ovo_

class ir

Base
osl
bas
Obj

as_dict

Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is

JSO
serial

class ir

Base
osl
bas
Ver

registr

class ir

Base
osl
bas
Ver

OBJ_BAS

alias
of
iron
obj
bas
Iro

serializ

Se-
ri-
al-
ize
the
en-
tity.

This
se-
ri-
al-
izes
the
en-
tity
so
that
it
can
be
sent
over

alized entity for an IronicObject is a dictionary with keys: `ironic_object.namespace`, `ironic_object.data`, `ironic_object.name`, `ironic_object.version`, and `ironic_object.changes`.

running the same or a newer release than the client. The client doesn't need to downgrade any IronicObjects when sending them over RPC. The server, on the other hand, will need to do so if the server is pinned and the target version of an IronicObject is older than the latest version of that Object.

e.g.
RPC
A
se-
ri-

We
as-
sum
that
the
client
(iron
API
is
al-
ways
talk-
ing
to
a
serv
(iron
conc
that
is

(In-
ter-
nally
the
ser-
vice
deal
with
the
lat-
est
ver-
sion
of
ob-
jects
so
we
know
that

these objects are always in the latest versions.)

Parameters

- **context**
security context
- **entity**
the entity to be serialized
may be an IronObject

Returns

the serialized entity

Raises

OverflowError (via .get_ironic_obj)

ironic.Object

Return the maximum number

ver-
sion
in
the
list.

Parameter

ver:

a
list
of
(stri
ver-
sion
as-
sum
to
have
at
least
one
en-
try

Returns

the
max
i-
mun
ver-
sion
(stri

ironic.objects.bios module

class ir

Base
ironic
objects
base
Ironic

VERSION

property

property

create(

Cre-
ate
a
BIO
Set-
ting
reco
in
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

Raises

Nod
Not-
Foun
if
the
node
id
is
not
foun

Raises

BIO
Set-
tin-
gAl-
read

ists
if
the
set-
ting
reco
al-
read
ex-
ists.

property

dbapi =

classmet

Dele
a
BIO
Set-
ting
base
on
its
node
and
nam

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
node
id.
- **nam**
BIO
set-
ting
nam
to

be
dele

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
the
bios
set-
ting
nam
is
not
foun

```
fields =
'oslo_ve
'attribu
'oslo_ve
'create
'oslo_ve
'lower_b
'oslo_ve
'max_len
'oslo_ve
'min_len
'oslo_ve
String(
'oslo_ve
'node_id
'oslo_ve
'read_on
'oslo_ve
'reset_r
'oslo_ve
'unique
'oslo_ve
'update
'oslo_ve
'upper_b
'oslo_ve
'value'
'oslo_ve
```

classme

Get
a
BIO
Set-
ting
base
on
its
node
and
nam

Parame

- **con**
Se-
cu-
rity
con-
text.

- **node**
The
node
id.

- **name**
BIO
set-
ting
name
to
be
re-
triev

Raises
Nod
Not-
Foun
if
the
node
id
is
not
foun

Raises
BIO
Set-
ting
Not-
Foun
if
the
bios
set-
ting
name
is
not
foun

Returns
A
:clas
ob-
ject.

property

property

property

property

property

property

registry

'max_le

'upper_l

property

save(con

Save

BIO

Set-

ting

up-

date

in

DB.

Parame

con

Se-

cu-

rity

con-

text.

NOT

This

shou

only

be

used

in-

ter-

nally

by

the

in-

di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
the
bios
set-
ting
nam
is
not
foun

property

property

property

property

class ir

Base
iro
obj
bas
Iro
iro
obj
bas
Iro

VERSION

classme

Cre-
ate
a
list
of
BIO
Set-
ting
reco
in
DB.

Paramet

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

- **nod**
The
node
id.

- **set**
A
list
of
bios
set-

tings

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco
al-
read
ex-
ists.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

property

dbapi =

classmet

Dele
BIO
Set-

tings
base
on
node
and
nam

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
node
id.
- **nam**
List
of
BIO
set-
ting
nam
to
be
dele

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting
Not-
Four

if
any
of
BIO
set-
ting
fails
to
dele

fields =
'oslo_ve
'objects
'oslo_ve
'updatee
'oslo_ve

classme

Get
BIO
Set-
ting
base
on
node

Parame

- **con**
Se-
cu-
rity
con-
text.

- **nod**
The
node
id.

Raises

Nod
Not-
Four
if
the
node
id
is
not

found

Returns

A list of BIOS Setting objects

property

classmethod

Save a list of BIOS Setting update in DB.

Parameter

- **context**
Security context. NOT This should only be used internally by the indi-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

- **node**
The
node
id.

- **set**
A
list
of
bios
set-
ting

Raises
Nod
Not-
Four
if
the
node
id
is
not
foun

Raises
BIO
Set-
ting-
Not-
Four
if
any
of
the
bios
set-
ting
nam
is
not
foun

Returns
A
list
of
BIO
Set-
ting
ob-

jects

classmethod

Return lists of create/update settings

This method syncs with bios data table and sorts out four lists of create/update settings

Parameters

- **context**
Security context.
- **nodeid**
The node id.
- **settings**
BIO settings

to
be
sync

Returns

A
4-
tuple
of
lists
of
BIO
set-
ting
to
be
cre-
ated
up-
date
dele
and
un-
char

property

ironic.objects.chassis module

class ir

Base
iro
obj
bas
Iro
osl
bas
Ver

VERSION

create(

Cre-
ate
a
Cha
sis
reco

in
the
DB.
Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.
If
tar-
get_
is
pro-
vide

it will be checked against the in-database copy of the chassis before updates are made.

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

property

dbapi =

property

destroy

Dele
the
Cha
sis
from
the
DB.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

property

fields =

'oslo_ve
'descrip
'oslo_ve
'extra'
'oslo_ve
Integer
'oslo_ve
'updatee
'oslo_ve
UUID(de
'oslo_ve

classmet

Find
a
chas
sis
base
on
its
id
or
uuid
and
re-
turn
a
Cha
sis
ob-
ject.

Parame

- **con**
Se-
cu-
rity
con-
text
- **cha**
the
id
or
uuid
of
a
chas
sis.

Returns

a
[Cha](#)
ob-
ject.

classmet

Find
a
chas

sis
base
on
its
in-
te-
ger
ID
and
re-
turn
a
Cha
sis
ob-
ject.

Parame

- **cls**
the
Cha
- **con**
Se-
cu-
rity
con-
text
- **cha**
the
ID
of
a
chas
sis.

Returns

a
Cha
ob-
ject.

classmet

Find
a
chas
sis

base
on
UI
and
re-
turn
a
Cha
ob-
ject.

Parame

- **cls**
the
Cha
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UI
of
a
chas
sis.

Returns

a
Cha
ob-
ject.

property

classme

Re-
turn
a
list
of
Cha
sis
ob-

jects

Parameters

- **cls**
the
Cha
- **con**
Se-
cu-
rity
con-
text.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.
- **sor**
col-
umn

to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

Returns
a
list
of
[Cha](#)
ob-
ject.

refresh

Loa
and
ap-
plies
up-
date
for
this
Cha
sis.

Loa
a
[Cha](#)
with
the
sam
uuid
from
the
data
and
chec
for
up-
date

applied from the loaded chassis column by column, if there are any updates.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

at-
tribu
Up-
date
are

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

save(con
Save
up-
date
to
this
Cha
sis.
Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the

re-
sult
of
self.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
whe
iron
cre-
ates,
up-

date
dele
a
chas
sis.

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci:
'warning
'oslo_ve
'publish
'oslo_ve
'updatec
'oslo_ve

property

property

property

property

class ir

Base
iro
obj
not
Not

SCHEMA =
('chass:
('chass:

VERSION

property

property

property

```
fields =  
'oslo_ve  
'descrip  
'oslo_ve  
'extra'  
'oslo_ve  
'updatec  
'oslo_ve  
UUID(de  
'oslo_ve
```

property

property

ironic.objects.conductor module

class ir

```
Base  
iro  
obj  
bas  
Iro  
oslo  
bas  
Ver
```

VERSION

property

property

dbapi =

property

```
fields =
'oslo_ve
'create
'oslo_ve
'driver
'oslo_ve
'hostname
'oslo_ve
Integer
'oslo_ve
'update
'oslo_ve
```

classme

Get
a
Con
duc-
tor
reco
by
its
host
nam

Parame

- **cls**
the
Con
- **con**
Se-
cu-
rity
con-
text
- **hos**
the
host
nam
on
whic
a
Con
duc-
tor
is

online field is ignored if this value is set to None.

run-
ning

- **onl.**
Spec
ify
the
ex-
pect
onl.
field
valu
for
the
con-
duc-
tor
to
be
re-
triev
The

Returns

a
Con
ob-
ject.

property

property

classme

Re-
turn
a
list
of
Con
duc-
tor
ob-
jects

Parame

- **cls**

the
[Con](#)

- **con**
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **sor**
col-
umn
to
sort
re-
sults
by.

-

sort
di-
rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
[Con](#)
ob-
ject.

refresh

Load
and
ap-
plies
up-
date
for
this
Con
duc-
tor.

Load
a
[Con](#)
with
the
same
uuid
from
the
data
and
check
for
up-
date
at-
tribu
Up-
date
are

applied from the loaded chassis column by column, if there are any updates.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `Conductor(context)`

Paramete
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

classmet

Reg-
is-
ter
an
ac-
tive
con-
duc-
tor
with
the
clus-
ter.

Paramete

- **cls**
the
Con
- **con**
Se-
cu-
rity
con-

text

- **hos**
the
host
nam
on
whic
the
con-
duc-
tor
will
run

- **dri**
the
list
of
driv
en-
able
in
the
con-
duc-
tor

- **con**
con-
duc-
tor
grou
to
join.
used
for
node
affin
ity.

- **upd**
Whe
false
reg-
is-
tra-
tion
will

line record is found. When true, will overwrite the existing record. Default: False.

raise
an
ex-
cep-
tion
when
a
con-
flict
ing
on-

Raises

Con
duc-
torA
read
is-
tere

Returns

a
Con
ob-
ject.

register

Reg-
is-
ter
hard
ware
in-
ter-
face
with
the
con-
duc-
tor.

Paramet

int
List
of
in-
ter-
face
to
reg-
is-

containing `hardware_type`, `interface_type`, `interface_name` and `default`, e.g. `{hardware_type: hardware-type, interface_type: deploy, interface_name: direct, default: True}`

ter,
each
en-
try
shou
be
a
dic-
tio-
nary

save(*con*

Save
is
not
sup-
port
by
Con
duc-
tor
ob-
jects

touch(*ca*

Touc
this
con-
duc-
tors
DB
reco
marl
ing
it
as
up-
to-
date

unregist

Re-
mov
this
con-
duc-
tor
from
the

ser-
vice
reg-
istry

unregist

Un-
reg-
is-
ter
all
hard
ware
in-
ter-
face
for
this
con-
duc-
tor.

property

ironic.objects.deploy_template module

class ir

Base
iro
obj
bas
Iro
osl
bas
Ver

VERSION

create(C

Cre-
ate
a
De-
ploy
plate
reco
in

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

the
DB.
Paramete
con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises
De-
ploy
plate
read
ists

if
a
de-
ploy
tem-
plate
with
the
same
UI
ex-
ists.

property

dbapi =

destroy

Dele
the
De-
ploy
plate
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

property

```
fields =  
'oslo_ve  
'extra'  
'oslo_ve  
Integer  
'oslo_ve  
'name':  
'oslo_ve  
'steps'  
'oslo_ve  
'updatec  
'oslo_ve  
UUID(de:  
'oslo_ve
```

classmet

Find
a
de-
ploy
tem-
plate
base
on
its
in-
te-
ger
ID.

Parame

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **tem**
The
ID
of
a
de-
ploy
tem-
plate

Raises
De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the

data

Returns

a
Deploy
ob-
ject.

classmethod

Find
a
de-
ploy
tem-
plate
base
on
its
nam

Parameter

- **context**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

- **name**
The
nam
of
a
de-

ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

Returns

a
Dep
ob-
ject.

classmet

Find
a
de-
ploy
tem-
plate
base
on
its
UUI

Paramet

- **con**
se-
cu-
rity
con-
text.
NOT
This

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `DeployTemplate(context)`.

shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **uui**
The
UUI
of
a
de-
ploy
tem-
plate

Raises
De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

Returns
a
Dep
ob-
ject.

property

classme

Re-
turn
a
list
of
De-
ploy
plate
ob-
jects

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a

single result.

- **marshal**
page information for large data sets.

- **sort**
column to sort results by.

- **sort**
direction to sort. asc or desc

Returns
a list of *Dep* objects

classmethod
Return a list of De-

ploy
plate
ob-
jects
matc
ing
a
set
of
nam

Paramete

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `DeployTemplate(context)`.

- **nam**
a
list
of
nam
to
fil-
ter
by.

Returns

a
list
of
Dep

ob-
jects

property

refresh

Loa
up-
date
for
this
de-
ploy
tem-
plate

Loa
a
de-
ploy
tem-
plate
with
the
sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu

Updates are applied from the loaded template column by column, if there are any updates.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

in-
ter-
nally
by
the
in-
di-

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

save(*con*

Save
up-
date
to
this
De-
ploy
plate
Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `DeployTemplate(context)`

self.
Parameters
context
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises
De-
ploy
plate
Four
if
the

de-
ploy
tem-
plate
does
not
ex-
ist.

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
on
de-
ploy
tem-
plate
API
op-
er-
a-
tions

VERSION

property

property

```
fields =  
'oslo_ve  
'event_t  
'oslo_ve  
'level'  
Unspeci  
'warning  
'oslo_ve  
'publisl  
'oslo_ve  
'updatec  
'oslo_ve
```

```
property
```

```
property
```

```
property
```

```
property
```

```
class ir
```

```
Base  
iro  
obj  
not  
Not
```

```
SCHEMA =  
('deploy  
'steps  
'updatec
```

```
VERSION
```

```
property
```

```
property
```

```
fields =
'oslo_ve
'extra'
'oslo_ve
String(
'oslo_ve
'steps'
'oslo_ve
'update
'oslo_ve
UUID(de
'oslo_ve
```

```
property
```

```
property
```

```
property
```

```
property
```

ironic.objects.deployment module

```
class ir
```

```
Base
iro
obj
bas
Iro
osl
bas
Ver
```

```
VERSION
```

```
create(
```

```
Cre-
ate
a
De-
ploy
men
Up-
date
the
```

cor-
re-
spor
ing
node
un-
der
the
hood

Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Deployment(context)

- **nod**
Nod
ob-
ject
for
de-
ploy
men

Raises

In-
stan
As-
so-
ci-
ated

Nod
As-
so-
ci-
ated
Nod
Not-
Four

property

dbapi =

destroy

Dele
the
De-
ploy
men

Up-
date
the
cor-
re-
spor
ing
node
un-
der
the
hoo

Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

nally
by
the
in-
di-

- **node**
Nod
ob-
ject
for
de-
ploy
men

```
fields =  
'oslo_ve  
'image_c  
'oslo_ve  
'image_r  
'oslo_ve  
'kernel_  
'oslo_ve  
'node_uu  
'oslo_ve  
'ramdisl  
'oslo_ve  
'root_de  
'oslo_ve  
'root_g  
'oslo_ve  
'state'  
'oslo_ve  
'swap_m  
'oslo_ve  
'updatee  
'oslo_ve  
UUID(de  
'oslo_ve
```

classmet

Find
a
de-
ploy
men
base
by
its

node
UI

Parameters

- **cls**
the
Dep
- **con**
Se-
cu-
rity
con-
text
- **node**
The
UI
of
a
cor-
re-
spon-
ding
node

Returns

An
Dep
ob-
ject.

Raises

Nod
Not-
Four

classmethod

Find
a
de-
ploy
men
by
its
UI

Parameters

- **cls**
the
Dep
- **con**
Se-
cu-
rity
con-
text
- **uui**
The
UUI
of
a
de-
ploy
men

Returns
An
Dep
ob-
ject.

Raises
In-
stan-
ceN
Four

property

property

instance
'image_3'
'ramdisl'
'swap_ml'

instance
'image_1'
'ramdisl'
'swap_m'

property

classme

Re-
turn
a
list
of
De-
ploy
men
ob-
jects

Parame

- **cls**
the
Dep
- **con**
Se-
cu-
rity
con-
text.
- **fil**
Fil-
ters
to
ap-
ply.
- **lim**
Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-

sult.

- **marshal**
Pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

Returns

A
list
of
[Dep](#)
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

node_map
'uuid':

property

property

refresh

Re-
fres
the
ob-
ject
by
re-
fetc
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

property

property

property

property

ionic.objects.fields module

property

property

class ir

Base
osl
fie
Boo

class ir

Base
osl
fie
Dat

class ir

Base
osl
fie
Enum

class ir

Base
osl
fie
Fie

static c

This
is
calle
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men

if this is not possible.

This
meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-
cep-
tion

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing
set

Param:

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-

ing
set

Returns

A
prop
type
valu

class ir

Base
osl
fie
Aut

AUTO_TYI

class ir

Base
osl
fie
Int

class ir

Base
osl
fie
Aut

AUTO_TYI

class ir

Base
osl
fie
Lis

class ir

Base
osl
fie
Lis

class ir

Base
osl
fie
Fie

static c

This
is
called
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men

This
meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-
cep-
tion

if this is not possible.

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-

tribu
is
be-
ing
set

Param::

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-
ing
set

Returns

A
prop
type
valu

class ir

Base
osl
fie
Aut

AUTO_TY

class ir

Base
osl
fie
Enum

ALL = (

CRITICAL

DEBUG =

ERROR =

INFO =

WARNING

class ir

Base
osl
fie
Bas

AUTO_TYI

class ir

Base
osl
fie
Enum

ALL = (

END = 'e

ERROR =

START =

SUCCESS

class ir

Base
osl
fie
Bas

AUTO_TYI

class ir

Base
osl
fie
Obj

class ir
Base
osl
fie
Str

static c

This
is
calle
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men

This
meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-
cep-
tion

if this is not possible.

Param:
The
Ver-
sion
dOb

ject
on
whic
an
at-
tribu
is
be-
ing
set

Param::

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-
ing
set

Returns

A
prop
type
valu

class ir

Base
osl
fie
Str

class ir

Base
osl
fie
Str

Cus-
tom
Strin
Field
ob-
ject

options, this StringField object allows for a function to be passed as a default, and will only process it at the point the field is coerced

ironic.objects.indirection module

that
al-
lows
for
func
tions
as
de-
fault
In
som
case
we
need
to
al-
low
for
dy-
nam
de-
fault
base
on
con-
fig-
u-
ra-
tion

AUTO_TYP

class ir

Base
osl
fie
UUI

class ir

Base
osl
bas
Ver

object_a

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
in-
stan

When
in-
di-
rec-
tion,
is
set
on
a
Ver-
sion
dOb
ject
(to
a
class
im-
ple-
men-
ing

this interface), method calls on remotable methods will cause this to be executed to actually make the desired call. This often involves performing RPC.

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the

ac-
tion

- **obj:**
The
ob-
ject
in-
stan-
ce
on
which
to
per-
form
the
ac-
tion

- **obj:**
The
name
of
the
ac-
tion
meth-
od
to
call

- **arg:**
The
po-
si-
tion
ar-
gu-
men-
t
to
the
ac-
tion
meth-
od

- **kwargs:**
The
key-
word
ar-
gu-

men
to
the
ac-
tion
meth

Returns

The
re-
sult
of
the
ac-
tion
meth

object_k

Per-
form
a
back
port
of
an
ob-
ject
in-
stan

This
meth
is
ba-
si-
cally
just
like
ob-
ject_
but
in-
stead
of
pro-
vid-
ing
a
spe-
cific

target version for the toplevel object and relying on the service-side mapping to handle sub-objects, this sends a mapping of all the dependent objects and their client-supported versions. The server will

backport objects within the tree starting at objinst to the versions specified in object_versions, removing objects that have no entry. Use obj_tree_get_versions() to generate this mapping.

plementedError if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to object_backport().

NOT
This
was
not
in
the
ini-
tial
spec
for
this
in-
ter-
face
so
the
base
class
raise
NotI

Parame

- **con**
The
con-
text
with
which
to
per-
form
the
back
port
- **obj**
An
in-
stan-
ce of
a
Ver-
sion
dOb

ject
to
be
back
port

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

object_

Dep
re-
cate
sinc
ver-
sion
0.10

Use
obj
in-
stead

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

Whe
in-
di-
rec-
tion
is
set

this interface), classmethod calls on `remotable_classmethod` methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

Parameters

- **context**
The context with which to perform the action
- **obj**
The registry name of the object
- **obj**
The name of the action method

to
call

- **obj**
The
(re-
mote
ver-
sion
of
the
ob-
ject
on
whic
the
ac-
tion
is
be-
ing
take

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

implementing VersionedObject class.

Returns

The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)
be
an
in-
stance
of
the

object_

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

When
in-
di-
rec-
tion,
is
set
on
a
Ver-
sion
dOb
ject
(to
a

this interface), classmethod calls on `remotable_classmethod` methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

client-side object versions for easier nested backports. The manifest is the result of calling `obj_tree_get_versions()`.

plementedError if you dont implement it. For backports, this method will be tried first, and if unim-

plemented, will fall back to `object_class_action()`. New implementations should provide this method instead of `object_class_action()`

Parameters

- **context**
The context with which to perform the action
- **objname**
The registry name of the object
- **objmethod**
The name of the action method to call
- **objversionmap**
A dict of {objname: version} map

ping

- **arg**
The
po-
si-
tion:
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs**
The
key-
word
ar-
gu-
men
to
the
ac-
tion
meth

Returns

The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)
be
an
in-
stan
of
the

implementing VersionedObject class.

ironic.objects.node module

class `ir`

Base
iro
obj
bas
Iro
osl
bas
Ver

VERSION

property

as_dict

Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is
JSO
seria

property

property

property

property

property

property

property

property

property

create()

Cre-
ate
a
Node
reco
in
the
DB.

Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.
If
tar-
get_
is
pro-
vide

it will be checked against the in-database copy of the node before updates are made.

Paramete
con
Se-
cu-
rity
con-
text.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
prop
erty
val-
ues
are
in-
valid

property

dbapi =

property

property

property

destroy

Dele
the
Nod
from

the
DB.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

property

property

property

property

property

fields =
'oslo_ve
'automat
'oslo_ve
'bios_in
'oslo_ve
'boot_in
'oslo_ve
'chassis
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'createe
'oslo_ve
'deploy.
'oslo_ve
'deploy.
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'driver.
'oslo_ve
'driver.
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
Integer
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten

classme

Find
a
node
base
on
its
id
or
uuid
and
re-
turn
a
Nod
ob-
ject.

Parame

- **con**
Se-
cu-
rity
con-
text
- **nod**
the
id
or
uuid
of
a
node

Returns

a
Nod
ob-
ject.

classme

Find
a
node
base
on

its
in-
te-
ger
ID
and
re-
turn
a
Node
ob-
ject.

Parameters

- **cls**
the
Node
- **context**
Se-
cu-
rity
con-
text
- **node_id**
the
ID
of
a
node

Returns

a
Node
ob-
ject.

classmethod

Find
a
node
base
on
the
in-
stan-
ce
UU

and
re-
turn
a
Nod
ob-
ject.

Parameters

- **cls**
the
Nod
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of
the
in-
stan

Returns

a
Nod
ob-
ject.

classmethod

Find
a
node
base
on
nam
and
re-
turn
a
Nod
ob-
ject.

Parameters

- **cls**
the [Node](#)
- **context**
Security context
- **name**
the logical name of a node

Returns

a [Node](#) object.

classmethod

Get a node by associated port address

Parameters

- **cls**
the [Node](#)

- **con**
Se-
cu-
rity
con-
text.

- **add**
A
list
of
port
ad-
dres

Raises
Nod
Not-
Foun
if
the
node
is
not
foun

Returns
a
Nod
ob-
ject.

classme
Find
a
node
base
on
UUI
and
re-
turn
a
Nod
ob-
ject.

Parame

-

cls
the
Nod

- **con**
Se-
cu-
rity
con-
text

- **uui**
the
UUI
of
a
node

Returns
a
Nod
ob-
ject.

property

property

property

property

property

property

property

property

classmet

Re-
turn
a
list
of

Nod
ob-
jects

Parame

- **cls**
the
Nod

- **con**
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **sor**

col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **fil**
Fil-
ters
to
ap-
ply.

Returns
a
list
of
Node
ob-
ject.

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

refresh

Re-
fresh
the
ob-
ject
by
re-
fetch
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

nally
by
the
in-
di-

classme

Re-
lease
the
rese
va-
tion
on
a
node

Parame

- **con**
Se-
cu-
rity
con-
text.
- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**
A
node
id
or
uuid

Raises

Node
Not-
Four
if
the
node
is
not
found

property

property

classmethod

Get
and
re-
serv
a
node

To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the
give
Node
whil
a
Task
is

performed, mark it reserved by this host.

Parameter

- **cls**
the

Nod

- **con**
Se-
cu-
rity
con-
text.

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold

- **nod**
A
node
ID
or
UUI

Raises
Nod
Not-
Foun
if
the
node
is
not
foun

Returns
a
Nod
ob-
ject.

property

property

property

save(*con*)

Save
up-
date
to
this
Nod

Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.
If
tar-
get_
is
pro-
vide

it will be checked against the in-database copy of the node before updates are made.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

di-

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

som

prop

erty

val-

ues

are

in-

valid

property

property

property

property

touch_p

Touc

the

data

reco

to

marl

the

pro-

vi-

sion

ing

as

alive

property

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
whe
iron
cre-
ates,
up-
date
or
dele
a
node

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci
'warning
'oslo_ve
'publisl
'oslo_ve
'updatee
'oslo_ve

property

property

property

property

class ir

Base

iro

obj

nod

Nod

Pay-

load

sche

for

wher

iron

cre-

ates,

up-

date

or

dele

a

node

```
SCHEMA =
('node',
'conduct',
('node',
'console',
'deploy',
('node',
'driver',
'extra',
'inspect',
'inspect',
'inspect',
'instance',
'instance',
('node',
'maintenance',
'manager',
('node',
'owner',
'power_',
('node',
'protect',
('node',
'provision',
'rescue',
('node',
'retired',
('node',
'target',
'target',
('node',
```

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property


```
fields =
'oslo_ve
'boot_in
'oslo_ve
'chassi:
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'createc
'oslo_ve
'deploy_
'oslo_ve
'deploy_
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'driver_
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten
'oslo_ve
'manager
'oslo_ve
String(
'oslo_ve
'network
'oslo_ve
'owner'
'oslo_ve
'power
```

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class ir

Base

iron

obj

not

Not

No-

ti-

fi-

ca-

tion

emit

ted

wher

node

con-

sole

state

char

VERSION

property

property

fields =

'oslo_ve

'event_t

'oslo_ve

'level'

Unspeci

'warning

'oslo_ve

'publisl

'oslo_ve

'updatee

'oslo_ve

property

property

property

property

class ir

Base

iron

obj

not

Not

No-

ti-

fi-

ca-

tion

for

wher

a

node

pow

state

is

cor-

a bare metal hardware is different from the power state on an ironic node (DB). This notification is emitted after the database is updated to reflect this correction.

recte
in
the
data

This
no-
ti-
fi-
ca-
tion
is
emit
ted
whe
iron
de-
tect
that
the
ac-
tual
pow
state
on

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci
'warning
'oslo_ve
'publish
'oslo_ve
'update
'oslo_ve

property

property

property

property

class ir

Base

iron

obj

nod

Nod

No-

ti-

fi-

ca-

tion

pay-

load

sche

for

whe

a

node

pow

state

is

cor-

recte

from

in-

di-

cate

the

pre-

vi-

ous

pow

state

on

the

iron

node

be-

fore

the

node

was

up-

dated.

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

```
fields =
'oslo_ve
'boot_in
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'create
'oslo_ve
'deploy.
'oslo_ve
'deploy.
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
'from_po
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten
'oslo_ve
'manager
'oslo_ve
String(
'oslo_ve
'network
'oslo_ve
'owner'
'oslo_ve
'power_i
'oslo_ve
'power_s
'oslo_ve
'proper
```


property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class ir

Base

iron

obj

not

Not

No-

ti-

fi-

ca-

tion

emit

ted

wher

main

te-

nanc

state

char

via

API

VERSION

property

property

fields =

'oslo_ve

'event_t

'oslo_ve

'level'

Unspeci:

'warning

'oslo_ve

'publish

'oslo_ve

'updatee

'oslo_ve

property

property

property

property

class ir

Base

iron

obj

not

Not

Base

class

used

for

all

no-

ti-

fi-

ca-

tion

pay-

load
about
a
Node
ob-
ject.

SCHEMA =
(**'node'** ,
'conduct'
(**'node'** ,
'console'
'deploy'
(**'node'** ,
'driver'
(**'node'** ,
'inspect'
'inspect'
'instanc
'last_er
'mainten
'manager'
(**'node'** ,
'owner'
'power_
(**'node'** ,
'protect'
(**'node'** ,
'provis
'rescue_
(**'node'** ,
'retirec
(**'node'** ,
'target_
'target_
(**'node'** ,

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

fields =
'oslo_ve
'boot_in
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'create
'oslo_ve
'deploy.
'oslo_ve
'deploy.
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten
'oslo_ve
'manager
'oslo_ve
String(
'oslo_ve
'network
'oslo_ve
'owner'
'oslo_ve
'power_
'oslo_ve
'power_s
'oslo_ve
'proper
'oslo_ve
'protec

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class ir

Base
iron
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
whe
iron
char
a
node
pow
state

VERSION

property

property

fields =

'oslo_ve

'event_t

'oslo_ve

'level'

Unspeci

'warning

'oslo_ve

'publisl

'oslo_ve

'updatee

'oslo_ve

property

property

property

property

class ir

Base

iro

obj

nod

Nod

Pay-

load

sche

for

wh

iron

chan

a

node

pow

state

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

fields =
'oslo_ve
'boot_in
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'create
'oslo_ve
'deploy.
'oslo_ve
'deploy.
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten
'oslo_ve
'manager
'oslo_ve
String(
'oslo_ve
'network
'oslo_ve
'owner'
'oslo_ve
'power_
'oslo_ve
'power_s
'oslo_ve
'proper
'oslo_ve
'protec

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
whe
iron
char
a
node
pro-
vi-

sion
state

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci:
'warning
'oslo_ve
'publish
'oslo_ve
'updatee
'oslo_ve

property

property

property

property

class ir

Base
iro
obj
nod
Nod

Pay-
load
sche
for
whe
iron
char
a
node

pro-
vi-
sion
state

SCHEMA =
('node' ,
'conduct
('node' ,
'console
'deploy.
('node' ,
'driver
'driver.
'fault')
'inspect
'inspect
'instanc
'instanc
('node' ,
'mainten
'managere
('node' ,
'owner' ,
'power_
('node' ,
'protect
('node' ,
'provis
'rescue.
('node' ,
'retire
('node' ,
'target.
'target.
('node' ,

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property


```
fields =
'oslo_ve
'boot_in
'oslo_ve
'clean_s
'oslo_ve
'conduct
'oslo_ve
'console
'oslo_ve
'console
'oslo_ve
'create
'oslo_ve
'deploy.
'oslo_ve
'deploy.
'oslo_ve
'descrip
'oslo_ve
'driver
'oslo_ve
'driver.
'oslo_ve
'event'
'oslo_ve
'extra'
'oslo_ve
'fault'
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'inspect
'oslo_ve
'instanc
'oslo_ve
'instanc
'oslo_ve
'last_er
'oslo_ve
'lessee
'oslo_ve
'mainten
'oslo_ve
'mainten
'oslo_ve
'manager
'oslo_ve
String(
'oslo_ve
'network
'oslo_ve
'owner'
'oslo_ve
'power
```

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

ironic.objects.notification module

class ir

Base
iron
obj
bas
Iro

De-
fines
the
even
to
be

tion being taken on the notification, and the status of the action.

sent
on
the
wire

An
Ever
Type
mus
spec
ify
the
ob-
ject
be-
ing
acte
on,
a
strin
de-
scrib
ing
the
ac-

VERSION

property

property

```
fields =  
'oslo_ve  
'create  
'oslo_ve  
'object  
'oslo_ve  
'status  
fields.U  
'error'  
'oslo_ve
```

property

property

to_event

Con
struc
strin
for
even
to
be
sent
on
the
wire

The
strin
is
in
the
for-
mat:
bare

Raises

Val-
ueEr
ror
if
self.
is
not
one
of
fie
Not.

Returns

even
strin

property

class ir

Base
iro
obj
bas
Iro

Base
class
for
ver-
sion

tionPayloadBase.

no-
ti-
fi-
ca-
tion:
Sub-
class:
must
de-
fine
the
pay-
load
field
which
must
be
a
sub-
class
of
No-
ti-
fi-
ca-

VERSION

property

emit (*con*

Send
the
no-
ti-
fi-
ca-
tion.

Raises

No-
ti-
fi-
ca-
tion:
Pay-
load
Er-
ror

Raises

oslo

property

fields =

'oslo_ve

'event_t

'oslo_ve

'level'

Unspeci

'warning

'oslo_ve

'updatee

'oslo_ve

property

property

property

class ir

Base

iro

obj

bas

Iro

Base

class

for

the

pay-

load

of

ver-

sion

no-

ti-

fi-

ca-

tions

SCHEMA =

VERSION

property

fields =

'oslo_ve

'updated

'oslo_ve

populat

Pop-

u-

late

the

ob-

ject

base

on

the

SCH

and

the

sour

ob-

jects

Parame

kwargs

A

dict

con-

tains

the

sour

ob-

ject

and

the

keys

de-

fine

in

the

SCH

Raises

No-

ti-

fi-

ca-

tion-

Sche

ject

ror

Raises

No-

ti-

fi-

ca-

tion-

Sche

Er-

ror

property

class `ironic`

Base

ironic

obj

bas

Iron

VERSION

property

fields =

'oslo_ve

String(

'oslo_ve

'service

'oslo_ve

'updatec

'oslo_ve

property

property

property

ironic.o

Re-

mov

se-

crets

from

pay-

ironic.objects.port module

load
ob-
ject.

class ir

Base
iro
obj
bas
Iro
osl
bas
Ver

VERSION

property

create(C

Cre-
ate
a
Port
reco
in
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

in-
di-

Raises

MA
read
ists
if
ad-
dres
col-
umn
is
not
uniq

Raises

Por-
tAl-
read
ists
if
uuid
col-
umn
is
not
uniq

property

dbapi =

destroy

Dele
the
Port
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Port
Not-
Four

property

```
fields =  
'oslo_ve  
'create  
'oslo_ve  
'extra'  
'oslo_ve  
Integer  
'oslo_ve  
'interna  
'oslo_ve  
'is_smar  
'local_  
'oslo_ve  
String(  
'oslo_ve  
'node_id  
'oslo_ve  
'physica  
'oslo_ve  
'portgro  
'oslo_ve  
'pxe_ena  
'oslo_ve  
'update  
'oslo_ve  
UUID(de  
'oslo_ve
```

classmet

object.

Find
a
port
Find
a
port
base
on
its
id
or
uuid
or
nam
or
MA
ad-
dres
and
re-
turn
a
Port

Parame

- **con**
Se-
cu-
rity
con-
text
- **por**
the
id
or
uuid
or
nam
or
MA
ad-
dres
of
a
port

Returns

a
Por
ob-
ject.

Raises

In-
va-
li-
dI-
den-
tity

classme

Find
a
port
base
on
ad-
dres
and
re-
turn
a
Por
ob-
ject.

Paramet

- **cls**
the
Por
- **con**
Se-
cu-
rity
con-
text
- **add**
the
ad-
dres
of
a

port

- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai

- **pro**
a
node
own
or
lesse
to
mat
agai

Returns

a
Por
ob-
ject.

Raises

Port
Not-
Four

classmet

Find
a
port
base
on
its
in-
te-
ger
ID
and
re-
turn
a
Port
ob-

ject.

Parameters

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **por**
the
ID
of
a
port

Returns

a
Port
ob-
ject.

Raises

Port
Not-
Four

classmethod

Find
a
port
base
on
nam
and
re-
turn
a
Port
ob-
ject.

Parameters

- **cls**
the
Por

- **con**
Se-
cu-
rity
con-
text

- **nam**
the
nam
of
a
port

Returns
a
Por
ob-
ject.

Raises
Port
Not-
Four

classmet
Find
a
port
base
on
UI
and
re-
turn
a
Por
ob-
ject.

Parame

- **cls**
the
Por

- **con**
Se-
cu-
rity
con-
text

- **uui**
the
UUI
of
a
port

Returns
a
Port
ob-
ject.

Raises
Port
Not-
Four

property

property

property

classme

Re-
turn
a
list
of
Port
ob-
jects

Paramet

- **con**
Se-
cu-
rity

con-
text.

- **limit**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **marshal**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc

or
desc

- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai

- **pro**
a
node
own
or
lesse
to
mat
agai

Returns
a
list
of
Port
ob-
ject.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

classmet

Re-
turn
a
list
of
Port
ob-
jects

as-
so-
ci-
ated
with
a
give
node
ID.

Parame

- **con**
Se-
cu-
rity
con-
text.

- **nod**
the
ID
of
the
node

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-

tion
marl
for
large
data
sets.

- **sort**
col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai

- **pro**
a
node
own
or
lesse
to
mat
agai

Returns
a
list

of
Port
ob-
ject.

classmet

Re-
turn
a
list
of
Port
ob-
jects
as-
so-
ci-
ated
with
a
give
port
group
ID.

Parame

- **con**
Se-
cu-
rity
con-
text.
- **por**
the
ID
of
the
port
group
- **lim**
max
i-
mun
num
ber

of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **sor**
col-
umn
to
sort
re-
sults
by.

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc

- **own**
DEF
RE-
CAT
a
node

own
to
mate
agai

- **pro.**
a
node
own
or
less
to
mate
agai

Returns

a
list
of
Por
ob-
ject.

property

property

property

property

property

property

refresh

Loa
up-
date
for
this
Port

Loa
a
port
with
the

applied from the loaded port column by column, if there are any updates.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu
Up-
date
are

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Port
Not-
Four

save(*con*)

Save
up-
date
to
this
Port

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Raises

Port
Not-
Four

Raises

MA
read
ists
if
ad-
dres

col-
umn
is
not
uniq

classmet

Re-
turn
when
is_s
field
is
sup-
port

Returns

When
is_s
field
is
sup-
port

Raises

ovo_

classmet

Re-
turn
when
the
phys
i-
cal_
field
is
sup-
port

Returns

When
the
phys
i-
cal_
field
is
sup-
port

Raises

ovo_

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
whe
iron
cre-
ates,
up-
date
or
dele
a
port

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci
'warning
'oslo_ve
'publisl
'oslo_ve
'updatee
'oslo_ve

property

property

property

property

class in

Base

iro

obj

not

Not

SCHEMA =

'created'

'is_smar

'local_

('port'

'updated'

VERSION

property

property

property

```
fields =
'oslo_ve
'create
'oslo_ve
'extra'
'oslo_ve
'is_smar
'local_
'oslo_ve
String(
'oslo_ve
'node_u
'oslo_ve
'physica
'oslo_ve
'portgre
'oslo_ve
'pxe_ena
'oslo_ve
'updatee
'oslo_ve
UUID(de
'oslo_ve

property

property

property

property

property

property

property

property
```

ironic.objects.portgroup module

class `ir`

Base
iro
obj
bas
Iro
osl
bas
Ver

VERSION

property

create()

Cre-
ate
a
Port
group
reco
in
the
DB.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A

context should be set when instantiating the object, e.g.: `Portgroup(context)`

Raises

Duplicate
Name
MA
read
ists,
Port
grou
read
ists

property

dbapi =

destroy

Dele
the
Port
grou
from
the
DB.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `Portgroup(context)`

Raises

Port
grou
Notl
Port
grou
Not-
Foun

property

fields =

'oslo_ve
'createc
'oslo_ve
'extra'
'oslo_ve
Integer
'oslo_ve
'intern
'oslo_ve
String(
'oslo_ve
String(
'oslo_ve
'node_id
'oslo_ve
'propert
'oslo_ve
'standa
'oslo_ve
'updatec
'oslo_ve
UUID(de
'oslo_ve

classme

Find
a
port
grou
base
on
its
id,
uuid
nam
or
ad-
dres

Parameters

- **port**
The id, uuid, name, or address of a port group.
- **context**
Security context.

Returns

A *Port* object.

Raises

InvalidIdentity

classmethod

Find port group by address and return a *Port* object.

Parameter

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **add**
The
MA
ad-
dres
of
a
port
grou
- **pro**
a
node
own
or
lesse
to
matc
agai

Returns

A
Port
ob-
ject.

Raises

Port
grou
Not-
Four

classmethod

Find
a
port

group
by
its
in-
te-
ger
ID
and
re-
turn
a
Port
group
ob-
ject.

Parameters

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **por**
The
ID
of
a
port
group

Returns

A
Port
ob-
ject.

Raises

Port
group
Not-
Four

classmet

Find
port
group
base
on
nam
and
re-
turn
a
Por
ob-
ject.

Parame

- **cls**
the
Por
- **con**
Se-
cu-
rity
con-
text
- **nam**
The
nam
of
a
port
grou

Returns

A
Por
ob-
ject.

Raises

Port
grou
Not-
Four

classme

Find

a
port
grou
by
UUI
and
re-
turn
a
Por
ob-
ject.

Parame

- **cls**
the
Por
- **con**
Se-
cu-
rity
con-
text
- **uui**
The
UUI
of
a
port
grou

Returns

A
Por
ob-
ject.

Raises

Port
grou
Not-
Four

property

property

classme

Re-
turn
a
list
of
Port
grou
ob-
jects

Paramet

- **cls**
the
Por

- **con**
Se-
cu-
rity
con-
text.

- **lim**
Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
Pag-
i-
na-
tion

marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
a
node
own
or
lesse
to
mat
agai

Returns

A
list
of
Por
ob-
ject.

Raises

In-
valid
Pa-
ram
e-
ter-
Valu

classme

Re-
turn
a
list
of
Port
grou
ob-
jects
as-
so-
ci-
ated
with
a
give
node
ID.

Parame

- **cls**
the
Por
- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
ID
of
the
node
- **lim**
Max
i-
mun
num
ber
of
re-

sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **marl**
Pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
a
node
own
or
lesse
to
mat

agai

Returns

A
list
of
Por
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

property

property

property

property

refresh

Loa
up-
date
for
this
Port
grou

Loa
a
port
grou
with
the
sam
uuid
from
the
data
and
chec
for

are applied from the loaded portgroup column by column, if there are any updates.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

up-
date
at-
tribu
Up-
date

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Port
grou
Not-
Four

save(con

Save
up-
date
to
this
Port
grou
Up-
date
will
be
mad
col-

umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

Raises

Port
grou
Not-
Four
Du-
pli-
cate
Nam
MA
read
ists

property

property

property

class ir

Base
iron
obj
not
Not

No-
ti-
fi-
ca-
tion
whe
iron
cre-
ates,
up-
date
or
dele
a
port
grou

VERSION

property

property

fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci
'warning
'oslo_ve
'publish
'oslo_ve
'update
'oslo_ve

property

property

property

property

class ir

Base

iro

obj

not

Not

SCHEMA =

('portgr

('portgr

('portgr

'standa

'uuid':

VERSION

property

property

property

fields =

'oslo_ve

'create

'oslo_ve

'extra'

'oslo_ve

String(

'oslo_ve

String(

'oslo_ve

'node_u

'oslo_ve

'propert

'oslo_ve

'standa

'oslo_ve

'update

'oslo_ve

UUID(de

'oslo_ve

property

property

property

property

property

property

property

ironic.objects.trait module

class ir

Base
iron
obj
bas
Iro

VERSION

create(

Cre-
ate
a
Trai
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

property

dbapi =

classmet

Dele
the
Trai
from
the
DB.

Paramet

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

- **nod**
The
id
of
a
node

- **tra**
A
trait
strin

Raises
Nod

Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

Raises

Nod
Trai
Not-
Four
if
the
trait
is
not
foun

classmeth

Che
whe
a
Trai
ex-
ists
in
the
DB.

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

in-
ter-
nally
by
the
in-
di-

- **node**
The
id
of
a
node

- **traits**
A
trait
string

Returns
True
if
the
trait
ex-
ists
oth-
er-
wise
False

Raises
Node
Not-
Found
if
the
node
no
long
ap-
pear
in
the
data

```
fields =  
'oslo_ve  
'node_id  
'oslo_ve  
'trait'  
'oslo_ve  
'updatec  
'oslo_ve
```

property

property

property

class ir

```
Base  
iron  
obj  
bas  
Iron  
iron  
obj  
bas  
Iron
```

VERSION

classme

Re-
plac
all
ex-
ist-
ing
trait
with
the
spec
i-
fied
list.

Parame

- con

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **node**
The
id
of
a
node

- **traits**
List
of
Strin
trait
to
set.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-
ceed

the
per-
node
trait
limi

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

property

dbapi =

classmet

Dele
all
trait
for
the
spec
i-
fied
node

Paramet

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

used
in-
ter-
nally
by
the
in-
di-

- **node**
The
id
of
a
node

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

```
fields =  
'oslo_ve  
'objects'  
'oslo_ve  
'updatee  
'oslo_ve
```

classmet

Re-
turn
all
trait
for
the
spec
i-
fied
node

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

Parameter

- **context**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **node**
The
id
of
a
node

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

get_trait

Re-
turn

a
list
of
nam
of
the
trait
in
this
list.

property

property

ironic.objects.volume_connector module

class `ironic.objects.volume_connector`

Base
ironic.objects.volume_connector
base
Ironic
oslo
base
Ver

VERSION

property

create()

Cre-
ate
a
Vol-
ume
Con-
nec-
tor
reco
in
the
DB.

Paramet
con

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

ready exists with the same type and connector_id

se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Vol-
ume
Con-
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
a
vol-
ume
con-
nec-
tor
al-

Raises

Vol-
ume
Con-
nec-
torA
read
ists
if

ready exists

a
vol-
ume
con-
nec-
tor
with
the
sam
UUI
al-

property

dbapi =

destroy

Dele
the
Vol-
ume
Con
nec-
tor
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A

context should be set when instantiating the object, e.g.: `VolumeConnector(context)`.

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

property

fields =
'oslo_ve
'created
'oslo_ve
'extra'
'oslo_ve
Integer
'oslo_ve
'node_id
'oslo_ve
String(
'oslo_ve
'update
'oslo_ve
UUID(de
'oslo_ve

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its

ID
or
UUID

Parameters

- **context**
security context
- **identifier**
the data primary key ID
or
the UUID of a volume connector

Returns

a *Volume* object

Raises

InvalidIdentity if identifier is neither an in-

fied ident

te-
ger
ID
nor
a
UUI

Raises

Vol-
ume
Con
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
in-
te-
ger
ID.

Parame

- **cls**
the
Vol

- **con**
Se-
cu-
rity
con-
text.

- **db_**
The
in-
te-
ger
(data
pri-
mar-
key)
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A
Vol
ob-
ject.

Raises

Vol-
ume
Con-
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-

fed ID.

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
UUI

Parame

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **uui**
the
UUI
of
a
vol-
ume
con-
nec-
tor

Returns

a
Vol
ob-
ject

Raises

Vol-
ume
Con
nec-

fixed UUID

torN
Four
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-

property

classme

Re-
turn
a
list
of
Vol-
ume
Con
nec-
tor
ob-
jects

Paramete

- **con**
se-
cu-
rity
con-
text
- **lim**
max
i-
mun
num
ber
of

re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **marl**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
The
as-
so-
ci-
ated
node

proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

classmet

Re-
turn
a
list
of
Vol-
ume
Con
nec-
tor
ob-
jects
re-
latec
to
a
give
node
ID.

Parame

•

con
se-
cu-
rity
con-
text

- **nod**
the
in-
te-
ger
ID
of
the
node

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets

- **sor**
col-

um
to
sort
re-
sults
by

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-

Valu
if
sort
does
not
ex-
ist

property

refresh

Loa
up-
date
for
this
Vol-
ume
Con
nec-
tor.

Loa
a
vol-
ume
con-
nec-
tor
with
the
sam
UI
from
the
data
and
chec
for
up-
date
at-

tributes. If there are any updates, they are applied from the loaded volume connector, column by column.

Param
con
se-
cu-
rity
con-
text.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

save(*con*

Save
up-
date
to
this
Vol-
ume
Con
nec-
tor.

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
se-
cu-
rity
con-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
an-
othe
con-

exists with the same values for type and connector_id fields

nec-
tor
al-
read

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
UU
is
be-
ing
char

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
at
CRU
of
a
vol-
ume
con-
nec-

tor.

VERSION

property

property

fields =

'oslo_ve

'event_t

'oslo_ve

'level'

Unspeci:

'warning

'oslo_ve

'publish

'oslo_ve

'updatec

'oslo_ve

property

property

property

property

class ir

Base

iro

obj

not

Not

Pay-

load

sche

for

CRU

of

a

vol-

ume

con-

nec-

tor.

```
SCHEMA =  
( 'conne  
( 'conne  
( 'conne
```

```
VERSION
```

```
property
```

```
property
```

```
property
```

```
fields =  
'oslo_ve  
'create  
'oslo_ve  
'extra'  
'oslo_ve  
'node_u  
'oslo_ve  
'type':  
'oslo_ve  
'update  
'oslo_ve  
UUID(de  
'oslo_ve
```

```
property
```

```
property
```

```
property
```

```
property
```

ironic.objects.volume_target module

class `ironic.objects.volume_target`

Base
ironic.objects.volume_target
BaseTarget
Ironi
oslo
base
Ver

VERSION

property

create()

Cre-
ate
a
Vol-
ume
get
reco
in
the
DB.

Paramete

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

the same node ID and boot index

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

Raises

Vol-
ume
ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with
the
sam
UU
ex-
ists

property

dbapi =

destroy

Dele

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

the
Vol-
ume
get
from
the
DB.

Paramete
con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

property


```
fields =
'oslo_ve
'create
'oslo_ve
'extra'
'oslo_ve
Integer
'oslo_ve
'node_id
'oslo_ve
'proper
'oslo_ve
'update
'oslo_ve
UUID(de
'oslo_ve
'volume
'oslo_ve
'volume
'oslo_ve
```

classme

```
Find
a
vol-
ume
tar-
get
base
on
its
ID
or
UUID
```

Parame

- **con**
se-
cu-
rity
con-
text
- **ide**
the
data
pri-
mar

key
ID
or
the
UUID
of
a
vol-
ume
tar-
get

Returns

a
Vol
ob-
ject

Raises

In-
va-
li-
dI-
den-
tity
if
iden-
is
nei-
ther
an
in-
te-
ger
ID
nor
a
UUID

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

iden
ex-
ists

classme

Find
a
vol-
ume
tar-
get
base
on
its
data
ID.

Parame

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **db_**
the
data
pri-
mar
key
(in-
te-
ger)
ID
of
a
vol-
ume
tar-
get

Returns

a

Vol-
ob-
ject

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists

classmet

Find
a
vol-
ume
tar-
get
base
on
its
UUI

Parame

- **cls**
the
Vol-
- **con**
se-
cu-
rity
con-
text
- **uui**
the

UUI
of
a
vol-
ume
tar-
get

Returns

a
Vol
ob-
ject

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this
UUI
ex-
ists

property

classmethod

Re-
turn
a
list
of
Vol-
ume
get
ob-
jects

Parameter

- **con**
se-

cu-
rity
con-
text

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-
rec-
tion
to

sort.
asc
or
desc

- **pro.**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram
e-
ter
Valu
if
sort
does
not
ex-
ist

classme

Re-
turn
a

list
of
Vol-
ume
get
ob-
jects
re-
lated
to
a
give
node
ID.

Parame

- **con**
se-
cu-
rity
con-
text
- **nod**
the
in-
te-
ger
ID
of
the
node
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-

gle
re-
sult

- **marl**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of

Vol-
ob-
jects

Returns

a
list
of
Vol-
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

classme

Re-
turn
a
list
of
Vol-
ume
get
ob-
jects
re-
lated
to
a
give
vol-
ume
ID.

Paramete

- **con**
se-

cu-
rity
con-
text

- **vol**
the
UUI
of
the
vol-
ume

- **lim**
max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
in
a
sin-
gle
re-

sult

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets

- **sor**
col-
umn
to

sort
re-
sults
by

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

property

property

refresh

Loa
up-
date
for
this
Vol-
ume

get.
Loa
a
vol-
ume
tar-
get
with
the
sam
UUI
from
the
data
and
chec
for
up-
date
at-
tribu

If there are any updates, they are applied from the loaded volume target, column by column.

Parame
con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

Raises
Vol-
ume
get-
Not-
Four

if
the
vol-
ume
tar-
get
can-
not
be
foun

save(*con*

Save
up-
date
to
this
Vol-
ume
get.

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

the same node ID and boot index values

in-
ter-
nally
by
the
in-
di-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
UUI
is
be-
ing
char

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with

Raises

Vol-
ume
get-
Not-

Four
if
the
vol-
ume
tar-
get
can-
not
be
foun

property

property

property

property

class ir

Base
iro
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
at
CRU
of
a
vol-
ume
tar-
get.

VERSION

property

property

```
fields =
'oslo_ve
'event_t
'oslo_ve
'level'
Unspeci:
'warning
'oslo_ve
'publish
'oslo_ve
'updatee
'oslo_ve
```

property

property

property

property

class ir

Base
iro
obj
not
Not

```
SCHEMA =
('target'
('target'
('target'
('target'
```

VERSION

property

property

property

```
fields =
'oslo_ve
'create
'oslo_ve
'extra'
'oslo_ve
'node_u
'oslo_ve
'propert
'oslo_ve
'update
'oslo_ve
UUID(de
'oslo_ve
'volume_
'oslo_ve
'volume_
'oslo_ve
```

property

property

property

property

property

property

Module contents

ironic.o

Submodules

ironic.version module

Module contents

jobs and how to debug failures that may arise. To facilitate that, we have created the documentation below.

Jobs description

8.1.
the
Iron
ics
CI

Its
im-
por-
tant
to
un-
der-
stand
the
role
of
each
job
in
the
CI,
how
to
add
new

The
de-
scrip-
tion
of
each
jobs
that
runs
in
the
CI
when
you
sub-
mit
a

patc
for
open

stack/ironic is visible in *Table. OpenStack Ironic CI jobs description*.

Job name	Description
ironic-tox-unit-with-driver-libs-python3	Runs Ironic unit tests with the driver dependencies installed under Python3
ironic-standalone	Deploys Ironic in standalone mode and runs tempest tests that match the regex <i>ironic_standalone</i> .
ironic-tempest-functional-python3	Deploys Ironic in standalone mode and runs tempest functional tests that matches the regex <i>ironic_tempest_plugin.tests.api</i> under Python3
ironic-grenade	Deploys Ironic in a DevStack and runs upgrade for all enabled services.
ironic-grenade-dsvm-multinode-multitenant	Deploys Ironic in a multinode DevStack and runs upgrade for all enabled services.
ironic-tempest-ipa-partition-pxe_ipmitool	Deploys Ironic in DevStack under Python3, configured to use dib ramdisk partition image with <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploy 1 virtual baremetal.
ironic-tempest-partition-bios-redfish-pxe	Deploys Ironic in DevStack, configured to use dib ramdisk partition image with <i>pxe</i> boot and <i>redfish</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> , also deploys 1 virtual baremetal.
ironic-tempest-ipa-partition-uefi-pxe_ipmitool	Deploys Ironic in DevStack, configured to use dib ramdisk partition image with <i>uefi</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> , also deploys 1 virtual baremetal.
ironic-tempest-ipa-whole-disk-direct-tinyipa-multinode	Deploys Ironic in a multinode DevStack, configured to use a pre-build tinyipa ramdisk whole-disk image that is downloaded from a Swift temporary url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex (<i>ironic_tempest_plugin.tests.scenario test_schedule_to_all_nod</i>) and deploys 7 virtual baremetal.
ironic-tempest-ipa-whole-disk-bios-agent_ipmitool-tinyipa	Deploys Ironic in DevStack, configured to use a pre-build tinyipa ramdisk whole-disk image that is downloaded from a Swift temporary url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploys 1 virtual baremetal.
ironic-tempest-ipa-whole-disk-bios-agent_ipmitool-indirect	Deploys Ironic in DevStack, configured to use a pre-built dib ramdisk whole-disk image that is downloaded from http url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploys 1 virtual baremetal.

Adding a new Job

Are you familiar with Zuul?

fig and the Zuul Best Practices.

Where can I find the existing jobs?

that contains three files, whose function is described below.

Be-
fore
start
try-
ing
to
fig-
ure
out
how
Zuul
work
take
some
time
and
read
about
Zuul
Con

The
jobs
for
the
Iron
proj
are
de-
fine
un-
der
the
zuul
fold
in
the
root
di-
rec-
tory.

verted to Zuul v3 yet.

iron
jobs
Con
tains
the
con-
fig-
u-
ra-
tion
of
each
Iron
Job
con-
verte
to
Zuul
v3.

•

lega
iron
jobs
Con
tains
the
con-
fig-
u-
ra-
tion
of
each
Iron
Job
that
have
been
con-

•

proj
Con
tains
the
jobs
that
will
run
dur-

Create a new Job

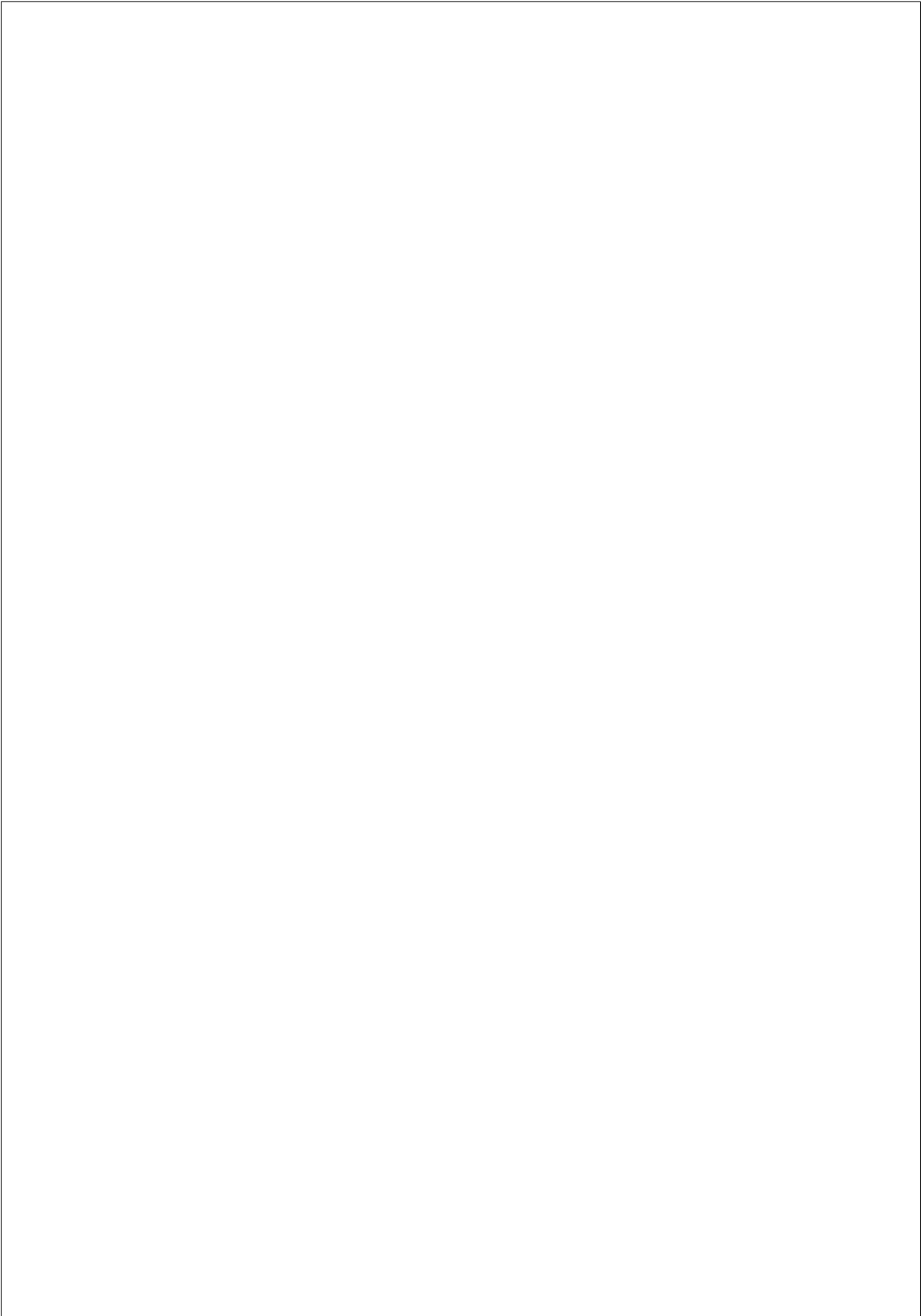
want to test, the existing job will be used as *parent* in your job definition. Now you will only need to either overwrite or add variables to your job definition under the *vars* section to represent the desired scenario.

ing
chec
and
gate
phas

Iden
tify
amo
the
ex-
ist-
ing
jobs
the
one
that
mos
clos
re-
sem
bles
the
sce-
nari
you

The
code
bloc
be-
low
show
the
min-
i-
mal
struc
ture
of
a
new
job
def-
i-
ni-

that you need to add to `ironic-jobs.yaml`.



(continues on next page)

you to find the initial reason for the failure. When clicking in the failed job you will be redirect to the Zuul web page that contains all the information about the job build.

Zuul Web Page

build failed it will contain a general output of the failure.

jobs
for
your
patch
please
don't
panic
This
guide
may
help

The
page
has
three
tabs
Sum
mar
Log
and
Con
sole

- Sum
mar
Con
tains
over
all
in-
for-
ma-
tion
about
the
build
of
the
job,
if
the
job

in the job. This will give you an overall idea of the failures and you can identify services that may be involved. The *job-output* file can give an overall idea of the failures and what services may be involved.

row before each playbook name you can find the roles and commands that were executed.

Log
Con
Con
tains
all
con-
fig-
u-
ra-
tion
and
log
files
about
all
ser-
vice
that
were
used

- Con
sole
Con
tains
all
the
play
book
that
were
ex-
e-
cute
by
click
ing
in
the
ar-

can be tagged as `Unmaintained`, after discussions within the ironic community. If such a decision is taken, an email will be sent to the OpenStack mailing list.

tively backport patches from maintained branches. Fixes can still be merged, though, if pushed into

review by operators or other downstream developers. It also means that branchless projects (e.g.: ironic-tempest-plugin), may not have configurations that are compatible with those branches.

As
of
09
Mar
2020
the
list
of
Unm
bran
in-
clud

- Oca
(Las
com
mit
-

Jun
28,
2019

- Pike
(Las
com
mit
-

Oct
2,
2019

CHAPTER

NINE

RELEASE NOTES

Release Notes

PYTHON MODULE INDEX

a

- `ironic.api`, 1081
- `ironic.api.app`, 1063
- `ironic.api.config`, 1064
- `ironic.api.controllers`, 1059
 - `ironic.api.controllers.base`, 1054
 - `ironic.api.controllers.link`, 1056
 - `ironic.api.controllers.root`, 1056
 - `ironic.api.controllers.v1`, 1053
 - `ironic.api.controllers.v1.allocation`, 805
 - `ironic.api.controllers.v1.bios`, 811
 - `ironic.api.controllers.v1.chassis`, 813
 - `ironic.api.controllers.v1.collection`, 819
 - `ironic.api.controllers.v1.conductor`, 822
 - `ironic.api.controllers.v1.deploy_template`, 826
 - `ironic.api.controllers.v1.driver`, 833
 - `ironic.api.controllers.v1.event`, 842
 - `ironic.api.controllers.v1.node`, 843
 - `ironic.api.controllers.v1.notification_utils`, 901
 - `ironic.api.controllers.v1.port`, 905
 - `ironic.api.controllers.v1.portgroup`, 916
 - `ironic.api.controllers.v1.ramdisk`, 924
 - `ironic.api.controllers.v1.utils`, 931
 - `ironic.api.controllers.v1.versions`, 1022
 - `ironic.api.controllers.v1.volume`, 1023
 - `ironic.api.controllers.v1.volume_connector`, 1023
 - `ironic.api.controllers.v1.volume_target`, 1038
 - `ironic.api.controllers.version`, 1057
- `ironic.api.functions`, 1064
- `ironic.api.hooks`, 1071
- `ironic.api.method`, 1079
- `ironic.api.middleware`, 1061

- `ironic.api.middleware.auth_public_routes`, 1059
- `ironic.api.middleware.json_ext`, 1060
- `ironic.api.middleware.parsable_error`, 1060
- `ironic.api.wsgi`, 1080

C

- `ironic.cmd`, 1084
 - `ironic.cmd.api`, 1081
 - `ironic.cmd.conductor`, 1081
 - `ironic.cmd.dbsync`, 1081
 - `ironic.cmd.status`, 1083
- `ironic.common`, 1372
 - `ironic.common.args`, 1093
 - `ironic.common.boot_devices`, 1110
 - `ironic.common.boot_modes`, 1113
 - `ironic.common.cinder`, 1114
 - `ironic.common.components`, 1128
 - `ironic.common.config`, 1130
 - `ironic.common.context`, 1130
 - `ironic.common.dhcp_factory`, 1132
 - `ironic.common.driver_factory`, 1136
 - `ironic.common.exception`, 1152
 - `ironic.common.faults`, 1176
 - `ironic.common.fsm`, 1177
 - `ironic.common.glance_service`, 1093
 - `ironic.common.glance_service.image_service`, 1084
 - `ironic.common.glance_service.service_utils`, 1091
 - `ironic.common.hash_ring`, 1183
 - `ironic.common.i18n`, 1183
 - `ironic.common.image_service`, 1184
 - `ironic.common.images`, 1197
 - `ironic.common.indicator_states`, 1216
 - `ironic.common.keystone`, 1216
 - `ironic.common.molds`, 1223
 - `ironic.common.network`, 1226
 - `ironic.common.neutron`, 1233
 - `ironic.common.nova`, 1260

- ironic.common.policy, 1262
- ironic.common.profiler, 1266
- ironic.common.pxe_utils, 1269
- ironic.common.raid, 1300
- ironic.common.release_mappings, 1307
- ironic.common.rpc, 1309
- ironic.common.rpc_service, 1313
- ironic.common.service, 1315
- ironic.common.states, 1315
- ironic.common.swift, 1334
- ironic.common.utils, 1342
- ironic.common.wsgi_service, 1370
- ironic.conductor, 1620
 - ironic.conductor.allocations, 1372
 - ironic.conductor.base_manager, 1377
 - ironic.conductor.cleaning, 1383
 - ironic.conductor.deployments, 1388
 - ironic.conductor.manager, 1393
 - ironic.conductor.notification_utils, 1412
 - ironic.conductor.rpcapi, 1418
 - ironic.conductor.steps, 1548
 - ironic.conductor.task_manager, 1554
 - ironic.conductor.utils, 1573
- ironic.conf, 1628
 - ironic.conf.agent, 1620
 - ironic.conf.anaconda, 1621
 - ironic.conf.ansible, 1621
 - ironic.conf.api, 1621
 - ironic.conf.audit, 1621
 - ironic.conf.auth, 1621
 - ironic.conf.cinder, 1622
 - ironic.conf.conductor, 1623
 - ironic.conf.console, 1623
 - ironic.conf.database, 1623
 - ironic.conf.default, 1623
 - ironic.conf.deploy, 1623
 - ironic.conf.dhcp, 1623
 - ironic.conf.drac, 1623
 - ironic.conf.glance, 1623
 - ironic.conf.healthcheck, 1624
 - ironic.conf.ibm, 1624
 - ironic.conf.ilo, 1624
 - ironic.conf.inspector, 1624
 - ironic.conf.ipmi, 1624
 - ironic.conf.irmc, 1624
 - ironic.conf.metrics, 1624
 - ironic.conf.metrics_statsd, 1624
 - ironic.conf.molds, 1625
 - ironic.conf.neutron, 1625
 - ironic.conf.nova, 1625

- ironic.conf.opts, 1625
- ironic.conf.pxe, 1627
- ironic.conf.redfish, 1627
- ironic.conf.service_catalog, 1627
- ironic.conf.snmp, 1627
- ironic.conf.swift, 1627
- ironic.conf.xclarity, 1628

d

- ironic.db, 1911
 - ironic.db.api, 1781
 - ironic.db.migration, 1911
 - ironic.db.sqlalchemy, 1781
 - ironic.db.sqlalchemy.api, 1628
 - ironic.db.sqlalchemy.migration, 1762
 - ironic.db.sqlalchemy.models, 1765
- ironic.dhcp, 1933
 - ironic.dhcp.base, 1911
 - ironic.dhcp.neutron, 1919
 - ironic.dhcp.none, 1926
- ironic.drivers, 3153
 - ironic.drivers.base, 2950
 - ironic.drivers.drac, 3110
 - ironic.drivers.fake_hardware, 3112
 - ironic.drivers.generic, 3116
 - ironic.drivers.hardware_type, 3120
 - ironic.drivers.ibm, 3125
 - ironic.drivers.ilo, 3126
 - ironic.drivers.intel_ipmi, 3130
 - ironic.drivers.ipmi, 3131
 - ironic.drivers.irmc, 3132
 - ironic.drivers.modules, 2950
 - ironic.drivers.modules.agent, 2523
 - ironic.drivers.modules.agent_base, 2553
 - ironic.drivers.modules.agent_client, 2590
 - ironic.drivers.modules.agent_power, 2625
 - ironic.drivers.modules.ansible, 1943
 - ironic.drivers.modules.ansible.deploy, 1933
 - ironic.drivers.modules.boot_mode_utils, 2634
 - ironic.drivers.modules.console_utils, 2643
 - ironic.drivers.modules.deploy_utils, 2650
 - ironic.drivers.modules.drac, 2063
 - ironic.drivers.modules.drac.bios, 1943
 - ironic.drivers.modules.drac.boot, 1967

[ironic.drivers.modules.drac.common](#), 1972
[ironic.drivers.modules.drac.inspect](#), 1975
[ironic.drivers.modules.drac.job](#), 1981
[ironic.drivers.modules.drac.management](#), 1985
[ironic.drivers.modules.drac.power](#), 2007
[ironic.drivers.modules.drac.raid](#), 2016
[ironic.drivers.modules.drac.utils](#), 2048
[ironic.drivers.modules.drac.vendor_passthru](#), 2051
[ironic.drivers.modules.fake](#), 2695
[ironic.drivers.modules.ibmc](#), 2106
[ironic.drivers.modules.ibmc.management](#), 2063
[ironic.drivers.modules.ibmc.mappings](#), 2079
[ironic.drivers.modules.ibmc.power](#), 2079
[ironic.drivers.modules.ibmc.raid](#), 2089
[ironic.drivers.modules.ibmc.utils](#), 2097
[ironic.drivers.modules.ibmc.vendor](#), 2099
[ironic.drivers.modules.ilo](#), 2259
[ironic.drivers.modules.ilo.bios](#), 2106
[ironic.drivers.modules.ilo.boot](#), 2112
[ironic.drivers.modules.ilo.common](#), 2149
[ironic.drivers.modules.ilo.console](#), 2192
[ironic.drivers.modules.ilo.firmware_processor](#), 2194
[ironic.drivers.modules.ilo.inspect](#), 2200
[ironic.drivers.modules.ilo.management](#), 2204
[ironic.drivers.modules.ilo.power](#), 2239
[ironic.drivers.modules.ilo.raid](#), 2247
[ironic.drivers.modules.ilo.vendor](#), 2254
[ironic.drivers.modules.image_cache](#), 2787
[ironic.drivers.modules.image_utils](#), 2793
[ironic.drivers.modules.inspect_utils](#), 2811
[ironic.drivers.modules.inspector](#), 2813
[ironic.drivers.modules.intel_ipmi](#), 2260
[ironic.drivers.modules.intel_ipmi.management](#), 2260
[ironic.drivers.modules.ipmitool](#), 2816
[ironic.drivers.modules.ipxe](#), 2862
[ironic.drivers.modules.irmc](#), 2330
[ironic.drivers.modules.irmc.bios](#), 2260
[ironic.drivers.modules.irmc.boot](#), 2267
[ironic.drivers.modules.irmc.common](#), 2283
[ironic.drivers.modules.irmc.inspect](#), 2291
[ironic.drivers.modules.irmc.management](#), 2297
[ironic.drivers.modules.irmc.power](#), 2316
[ironic.drivers.modules.irmc.raid](#), 2325
[ironic.drivers.modules.network](#), 2379
[ironic.drivers.modules.network.common](#), 2330
[ironic.drivers.modules.network.flat](#), 2348
[ironic.drivers.modules.network.neutron](#), 2358
[ironic.drivers.modules.network.noop](#), 2369
[ironic.drivers.modules.noop](#), 2862
[ironic.drivers.modules.noop_mgmt](#), 2887
[ironic.drivers.modules.pxe](#), 2901
[ironic.drivers.modules.pxe_base](#), 2914
[ironic.drivers.modules.redfish](#), 2490
[ironic.drivers.modules.redfish.bios](#), 2379
[ironic.drivers.modules.redfish.boot](#), 2388
[ironic.drivers.modules.redfish.inspect](#), 2403
[ironic.drivers.modules.redfish.management](#), 2408
[ironic.drivers.modules.redfish.power](#), 2448
[ironic.drivers.modules.redfish.raid](#), 2457
[ironic.drivers.modules.redfish.utils](#), 2479
[ironic.drivers.modules.redfish.vendor](#), 2486
[ironic.drivers.modules.snmp](#), 2924
[ironic.drivers.modules.storage](#), 2507
[ironic.drivers.modules.storage.cinder](#), 2490
[ironic.drivers.modules.storage.external](#), 2496
[ironic.drivers.modules.storage.noop](#), 2502
[ironic.drivers.modules.xclarity](#), 2523
[ironic.drivers.modules.xclarity.common](#), 2507

- `ironic.drivers.modules.xclarity.management`,
2512
- `ironic.drivers.modules.xclarity.power`,
2518
- `ironic.drivers.redfish`, 3135
- `ironic.drivers.snmp`, 3137
- `ironic.drivers.utils`, 3137
- `ironic.drivers.xclarity`, 3152

i

- `ironic`, 3407

O

- `ironic.objects`, 3406
- `ironic.objects.allocation`, 3153
- `ironic.objects.base`, 3168
- `ironic.objects.bios`, 3182
- `ironic.objects.chassis`, 3197
- `ironic.objects.conductor`, 3208
- `ironic.objects.deploy_template`, 3217
- `ironic.objects.deployment`, 3233
- `ironic.objects.fields`, 3242
- `ironic.objects.indirection`, 3250
- `ironic.objects.node`, 3262
- `ironic.objects.notification`, 3311
- `ironic.objects.port`, 3318
- `ironic.objects.portgroup`, 3340
- `ironic.objects.trait`, 3357
- `ironic.objects.volume_connector`, 3367
- `ironic.objects.volume_target`, 3386

V

- `ironic.version`, 3406

Symbols

--autogenerate
 revision command line option, 539
 --config-dir <DIR>
 ironic-dbsync command line option,
 537
 --config-file <PATH>
 ironic-dbsync command line option,
 537
 --debug
 ironic-dbsync command line option,
 537
 --help
 create_schema command line option,
 538
 ironic-dbsync command line option,
 537
 online_data_migrations command line
 option, 538
 revision command line option, 539
 stamp command line option, 539
 upgrade command line option, 540
 version command line option, 540
 --max-count <NUMBER>
 online_data_migrations command line
 option, 538
 --message <MESSAGE>
 revision command line option, 539
 --option <MIGRATION.KEY=VALUE>
 online_data_migrations command line
 option, 538
 --revision <REVISION>
 stamp command line option, 539
 upgrade command line option, 540
 --version
 ironic-dbsync command line option,
 537
 -d
 ironic-dbsync command line option,
 537
 -h

 create_schema command line option,
 538
 ironic-dbsync command line option,
 537
 online_data_migrations command line
 option, 538
 revision command line option, 539
 stamp command line option, 539
 upgrade command line option, 540
 version command line option, 540
 -m <MESSAGE>
 revision command line option, 539

A

abandon_bios_config()
 (*ironic.drivers.modules.drac.vendor_passthru.DracWSMa*
 method), 2053
 abandon_config() (in module
 ironic.drivers.modules.drac.bios), 1951
 abandon_config() (in module
 ironic.drivers.modules.drac.raid), 2030
 abort() (*ironic.drivers.base.InspectInterface*
 method), 3000
 abort() (*ironic.drivers.modules.inspector.Inspector*
 method), 2813
 abort_on_conductor_take_over() (in module
 ironic.conductor.utils), 1573
 AbstractHardwareType (class in
 ironic.drivers.hardware_type), 3120
 acquire() (in module
 ironic.conductor.task_manager), 1572
 acquire_port() (in module
 ironic.drivers.modules.console_utils),
 2643
 action (*ironic.objects.notification.EventType*
 property), 3312
 activate_license()
 (*ironic.drivers.modules.ilo.management.IloManagement*
 method), 2207
 ACTIVE (in module *ironic.common.states*), 1316
 add_allocation_filter_by_conductor() (in

- module ironic.db.sqlalchemy.api*), 1757
- `add_allocation_filter_by_node()` (in *module ironic.db.sqlalchemy.api*), 1757
- `add_auth_opts()` (in *module ironic.conf.auth*), 1621
- `add_certificates()` (in *module ironic.drivers.modules.ilo.common*), 2151
- `add_cleaning_network()` (*ironic.drivers.base.NetworkInterface* method), 3043
- `add_cleaning_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 2349
- `add_cleaning_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 2359
- `add_cleaning_network()` (*ironic.drivers.modules.network.noop.NoopNetwork* method), 2369
- `add_command_parsers()` (in *module ironic.cmd.dbsync*), 1083
- `add_fieldname()` (*ironic.common.exception.UnknownAttribute* method), 1173
- `add_identity_filter()` (in *module ironic.db.sqlalchemy.api*), 1757
- `add_inspection_network()` (*ironic.drivers.base.NetworkInterface* method), 3043
- `add_inspection_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 2350
- `add_inspection_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 2360
- `add_node_capability()` (in *module ironic.drivers.utils*), 3140
- `add_node_filter_by_chassis()` (in *module ironic.db.sqlalchemy.api*), 1758
- `add_node_tag()` (*ironic.db.api.Connection* method), 1782
- `add_node_tag()` (*ironic.db.sqlalchemy.api.Connection* method), 1628
- `add_node_trait()` (*ironic.db.api.Connection* method), 1783
- `add_node_trait()` (*ironic.db.sqlalchemy.api.Connection* method), 1629
- `add_node_traits()` (*ironic.conductor.manager.ConductorManager* method), 1396
- `add_node_traits()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1430
- `add_port_filter()` (in *module ironic.db.sqlalchemy.api*), 1758
- `add_port_filter_by_node()` (in *module ironic.db.sqlalchemy.api*), 1759
- `add_port_filter_by_node_owner()` (in *module ironic.db.sqlalchemy.api*), 1759
- `add_port_filter_by_node_project()` (in *module ironic.db.sqlalchemy.api*), 1760
- `add_port_filter_by_portgroup()` (in *module ironic.db.sqlalchemy.api*), 1760
- `add_portgroup_filter()` (in *module ironic.db.sqlalchemy.api*), 1760
- `add_portgroup_filter_by_node()` (in *module ironic.db.sqlalchemy.api*), 1761
- `add_portgroup_filter_by_node_project()` (in *module ironic.db.sqlalchemy.api*), 1761
- `add_ports_to_network()` (in *module ironic.common.neutron*), 1234
- `add_provisioning_network()` (*ironic.drivers.base.NetworkInterface* method), 3044
- `add_provisioning_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 2351
- `add_provisioning_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 2361
- `add_provisioning_network()` (*ironic.drivers.modules.network.noop.NoopNetwork* method), 2370
- `add_rescuing_network()` (*ironic.drivers.base.NetworkInterface* method), 3045
- `add_rescuing_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 2351
- `add_rescuing_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 2361
- `add_secret_token()` (in *module ironic.conductor.utils*), 1573
- `add_state()` (*ironic.common.fsm.FSM* method), 1178
- `add_transition()` (*ironic.common.fsm.FSM* method), 1179

[add_volume_conn_filter_by_node_project\(\)](#) (*in module `ironic.db.sqlalchemy.api`*), 1761
[add_volume_target_filter_by_node_project\(\)](#) (*in module `ironic.db.sqlalchemy.api`*), 1761
[address](#) (*ironic.db.sqlalchemy.models.Port attribute*), 1777
[address](#) (*ironic.db.sqlalchemy.models.Portgroup attribute*), 1778
[address](#) (*ironic.objects.port.Port property*), 3318
[address](#) (*ironic.objects.port.PortCRUDPayload property*), 3338
[address](#) (*ironic.objects.portgroup.Portgroup property*), 3340
[address](#) (*ironic.objects.portgroup.PortgroupCRUDPayload property*), 3356
[ADOPTFAIL](#) (*in module `ironic.common.states`*), 1316
[ADOPTING](#) (*in module `ironic.common.states`*), 1317
[advanced_net_fields](#) (*ironic.api.controllers.v1.port.PortsController attribute*), 905
[after\(\)](#) (*ironic.api.hooks.ContextHook method*), 1072
[after\(\)](#) (*ironic.api.hooks.NoExceptionTracebackHook method*), 1076
[agent_add_clean_params\(\)](#) (*in module `ironic.drivers.modules.deploy_utils`*), 2650
[agent_is_alive\(\)](#) (*in module `ironic.conductor.utils`*), 1574
[AgentAPIError](#), 1152
[AgentBaseMixin](#) (*class in `ironic.drivers.modules.agent_base`*), 2553
[AgentClient](#) (*class in `ironic.drivers.modules.agent_client`*), 2590
[AgentCommandTimeout](#), 1152
[AgentConnectionFailed](#), 1152
[AgentDeploy](#) (*class in `ironic.drivers.modules.agent`*), 2523
[AgentDeployMixin](#) (*class in `ironic.drivers.modules.agent_base`*), 2561
[AgentInProgress](#), 1153
[AgentOobStepsMixin](#) (*class in `ironic.drivers.modules.agent_base`*), 2574
[AgentPower](#) (*class in `ironic.drivers.modules.agent_power`*), 2625
[AgentRAID](#) (*class in `ironic.drivers.modules.agent`*), 2525
[AgentRescue](#) (*class in `ironic.drivers.modules.agent`*), 2533
[ALL](#) (*ironic.objects.fields.NotificationLevel attribute*), 3246
[ALL](#) (*ironic.objects.fields.NotificationStatus attribute*), 3247
[ALL_INTERFACES](#) (*in module `ironic.drivers.base`*), 2950
[all_interfaces](#) (*ironic.drivers.base.BareDriver property*), 2958
[all_interfaces\(\)](#) (*in module `ironic.common.driver_factory`*), 1137
[all_versions\(\)](#) (*in module `ironic.api.controllers.version`*), 1057
[Allocation](#) (*class in `ironic.objects.allocation`*), 3153
[allocation_id](#) (*ironic.db.sqlalchemy.models.Node attribute*), 1772
[allocation_id](#) (*ironic.objects.node.Node property*), 3262
[allocation_sanitize\(\)](#) (*in module `ironic.api.controllers.v1.allocation`*), 811
[AllocationAlreadyExists](#), 1153
[AllocationCRUDNotification](#) (*class in `ironic.objects.allocation`*), 3165
[AllocationCRUDPayload](#) (*class in `ironic.objects.allocation`*), 3166
[AllocationDuplicateName](#), 1153
[AllocationFailed](#), 1153
[AllocationNotFound](#), 1153
[AllocationsController](#) (*class in `ironic.api.controllers.v1.allocation`*), 805
[allow_agent_token\(\)](#) (*in module `ironic.api.controllers.v1.utils`*), 931
[allow_agent_version_in_heartbeat\(\)](#) (*in module `ironic.api.controllers.v1.utils`*), 931
[allow_allocation_backfill\(\)](#) (*in module `ironic.api.controllers.v1.utils`*), 932
[allow_allocation_owner\(\)](#) (*in module `ironic.api.controllers.v1.utils`*), 933
[allow_allocation_update\(\)](#) (*in module `ironic.api.controllers.v1.utils`*), 933

- `allow_allocations()` (in module `ironic.api.controllers.v1.utils`), 934
- `allow_bios_interface()` (in module `ironic.api.controllers.v1.utils`), 935
- `allow_build_configdrive()` (in module `ironic.api.controllers.v1.utils`), 935
- `allow_configdrive_vendor_data()` (in module `ironic.api.controllers.v1.utils`), 936
- `allow_deploy_steps()` (in module `ironic.api.controllers.v1.utils`), 937
- `allow_deploy_templates()` (in module `ironic.api.controllers.v1.utils`), 937
- `allow_detail_query()` (in module `ironic.api.controllers.v1.utils`), 937
- `allow_dynamic_drivers()` (in module `ironic.api.controllers.v1.utils`), 938
- `allow_dynamic_interfaces()` (in module `ironic.api.controllers.v1.utils`), 939
- `allow_expose_conductors()` (in module `ironic.api.controllers.v1.utils`), 939
- `allow_expose_events()` (in module `ironic.api.controllers.v1.utils`), 940
- `allow_field()` (in module `ironic.api.controllers.v1.utils`), 941
- `allow_inject_nmi()` (in module `ironic.api.controllers.v1.utils`), 941
- `allow_inspect_abort()` (in module `ironic.api.controllers.v1.utils`), 941
- `allow_inspect_wait_state()` (in module `ironic.api.controllers.v1.utils`), 942
- `allow_links_node_states_and_driver_properties()` (in module `ironic.api.controllers.v1.utils`), 943
- `allow_local_link_connection_network_type()` (in module `ironic.api.controllers.v1.utils`), 943
- `allow_node_logical_names()` (in module `ironic.api.controllers.v1.utils`), 943
- `allow_node_rebuild_with_configdrive()` (in module `ironic.api.controllers.v1.utils`), 944
- `allow_port_advanced_net_fields()` (in module `ironic.api.controllers.v1.utils`), 944
- `allow_port_internal_info()` (in module `ironic.api.controllers.v1.utils`), 945
- `allow_port_is_smartnic()` (in module `ironic.api.controllers.v1.utils`), 945
- `allow_port_physical_network()` (in module `ironic.api.controllers.v1.utils`), 946
- `allow_portgroup_mode_properties()` (in module `ironic.api.controllers.v1.utils`), 947
- `allow_portgroups()` (in module `ironic.api.controllers.v1.utils`), 947
- `allow_portgroups_subcontrollers()` (in module `ironic.api.controllers.v1.utils`), 948
- `allow_query_bios()` (in module `ironic.api.controllers.v1.utils`), 948
- `allow_raid_config()` (in module `ironic.api.controllers.v1.utils`), 949
- `allow_ramdisk_endpoints()` (in module `ironic.api.controllers.v1.utils`), 949
- `allow_remove_chassis_uuid()` (in module `ironic.api.controllers.v1.utils`), 950
- `allow_rescue_interface()` (in module `ironic.api.controllers.v1.utils`), 950
- `allow_reset_interfaces()` (in module `ironic.api.controllers.v1.utils`), 951
- `allow_soft_power_off()` (in module `ironic.api.controllers.v1.utils`), 951
- `allow_status_in_heartbeat()` (in module `ironic.api.controllers.v1.utils`), 952
- `allow_storage_interface()` (in module `ironic.api.controllers.v1.utils`), 952
- `allow_traits()` (in module `ironic.api.controllers.v1.utils`), 953
- `allow_verify_ca_in_heartbeat()` (in module `ironic.api.controllers.v1.utils`), 953
- `allow_vifs_subcontroller()` (in module `ironic.api.controllers.v1.utils`), 953
- `allow_volume()` (in module `ironic.api.controllers.v1.utils`), 954
- `allowable_values` (`ironic.db.sqlalchemy.models.BIOSSetting` attribute), 1767
- `allowable_values` (`ironic.objects.bios.BIOSSetting` property), 3182
- `and_valid()` (in module `ironic.common.args`), 1093
- `AnsibleDeploy` (class in `ironic.drivers.modules.ansible.deploy`), 1934
- `apply_configuration()` (`ironic.drivers.base.BIOSInterface` method), 2951
- `apply_configuration()` (`ironic.drivers.base.RAIDInterface` method), 3071
- `apply_configuration()`

(ironic.drivers.modules.agent.AgentRAID method), 2526
attach_volumes() (in module *ironic.common.cinder*), 1114
apply_configuration() (*ironic.drivers.modules.drac.bios.DracWSManBIOS method*), 1945
apply_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID method*), 2022
apply_configuration() (*ironic.drivers.modules.fake.FakeBIOS method*), 2696
apply_configuration() (*ironic.drivers.modules.ibm.raid.IbmcRAID method*), 2090
apply_configuration() (*ironic.drivers.modules.ilo.bios.IloBIOS method*), 2106
apply_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID method*), 2247
apply_configuration() (*ironic.drivers.modules.irmc.bios.IRMCBIOS method*), 2261
apply_configuration() (*ironic.drivers.modules.noop.NoBIOS method*), 2864
apply_configuration() (*ironic.drivers.modules.redfish.bios.RedfishBIOS method*), 2379
apply_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID method*), 2457
apply_jsonpatch() (in module *ironic.api.controllers.v1.utils*), 955
args (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 1771
arguments (*ironic.api.functions.FunctionDefinition* attribute), 1065
as_dict() (*ironic.db.sqlalchemy.models.IronicBase* method), 1772
as_dict() (*ironic.objects.base.IronicObject* method), 3169
as_dict() (*ironic.objects.base.IronicObjectListBase* method), 3178
as_dict() (*ironic.objects.node.Node* method), 3262
attach_boot_iso_if_needed() (in module *ironic.drivers.modules.irmc.boot*), 2280
attach_vmedia() (in module *ironic.drivers.modules.ilo.common*), 2153
attach_volumes() (*ironic.drivers.base.StorageInterface method*), 3088
attach_volumes() (*ironic.drivers.modules.fake.FakeStorage method*), 2776
attach_volumes() (*ironic.drivers.modules.storage.cinder.CinderStorage method*), 2490
attach_volumes() (*ironic.drivers.modules.storage.external.ExternalStorage method*), 2496
attach_volumes() (*ironic.drivers.modules.storage.noop.NoopStorage method*), 2502
attribute_type (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
attribute_type (*ironic.objects.bios.BIOSSetting* property), 3182
AUTH_CLASSES (*ironic.drivers.modules.redfish.utils.SessionCache* attribute), 2479
authorize() (in module *ironic.common.policy*), 1262
AuthPublicRoutes (class in *ironic.api.middleware*), 1061
AuthPublicRoutes (class in *ironic.api.middleware.auth_public_routes*), 1059
AUTO_TYPE (*ironic.objects.fields.FlexibleDictField* attribute), 3244
AUTO_TYPE (*ironic.objects.fields.ListOfFlexibleDictsField* attribute), 3244
AUTO_TYPE (*ironic.objects.fields.MACAddressField* attribute), 3246
AUTO_TYPE (*ironic.objects.fields.NotificationLevelField* attribute), 3247
AUTO_TYPE (*ironic.objects.fields.NotificationStatusField* attribute), 3247
AUTO_TYPE (*ironic.objects.fields.StringFieldThatAcceptsCallable* attribute), 3250
automated_clean (*ironic.db.sqlalchemy.models.Node* attribute), 1772
automated_clean (*ironic.objects.node.Node* property), 3262
AVAILABLE (in module *ironic.common.states*), 1317

B

- `backfill_allocation()` (in module `ironic.conductor.allocations`), 1372
- `backup_bios_config()` (in module `ironic.drivers.modules.irmc.management`), 2315
- `BadRequest`, 1154
- `BareDriver` (class in `ironic.drivers.base`), 2957
- `BaseConductorManager` (class in `ironic.conductor.base_manager`), 1378
- `BaseDHCP` (class in `ironic.dhcp.base`), 1912
- `BaseDriverFactory` (class in `ironic.common.driver_factory`), 1136
- `BaseImageService` (class in `ironic.common.image_service`), 1184
- `BaseInterface` (class in `ironic.drivers.base`), 2964
- `before()` (`ironic.api.hooks.ConfigHook` method), 1071
- `before()` (`ironic.api.hooks.ContextHook` method), 1073
- `before()` (`ironic.api.hooks.DBHook` method), 1074
- `before()` (`ironic.api.hooks.PublicUrlHook` method), 1077
- `before()` (`ironic.api.hooks.RPCHook` method), 1078
- `BIOS` (in module `ironic.common.boot_devices`), 1112
- `bios` (`ironic.drivers.base.BareDriver` attribute), 2958
- `bios_interface` (`ironic.db.sqlalchemy.models.Node` attribute), 1772
- `bios_interface` (`ironic.objects.node.Node` property), 3262
- `bios_interface` (`ironic.objects.node.NodeCorrectedPowerStatePayload` property), 3291
- `bios_interface` (`ironic.objects.node.NodeCRUDPayload` property), 3283
- `bios_interface` (`ironic.objects.node.NodePayload` property), 3296
- `bios_interface` (`ironic.objects.node.NodeSetPowerStatePayload` property), 3302
- `bios_interface` (`ironic.objects.node.NodeSetProvisionStatePayload` property), 3307
- `BIOSInterface` (class in `ironic.drivers.base`), 2951
- `BIOSSetting` (class in `ironic.db.sqlalchemy.models`), 1766
- `BIOSSetting` (class in `ironic.objects.bios`), 3182
- `BIOSSettingAlreadyExists`, 1153
- `BIOSSettingList` (class in `ironic.objects.bios`), 3189
- `BIOSSettingListNotFound`, 1153
- `BIOSSettingNotFound`, 1154
- `BLINKING` (in module `ironic.common.indicator_states`), 1216
- `bmc_reset()` (`ironic.drivers.modules.ipmitool.VendorPassthru` method), 2848
- `body()` (in module `ironic.api.method`), 1079
- `body_type` (`ironic.api.functions.FunctionDefinition` attribute), 1066
- `boolean()` (in module `ironic.common.args`), 1095
- `BooleanField` (class in `ironic.objects.fields`), 3242
- `boot` (`ironic.drivers.base.BareDriver` attribute), 2958
- `boot_device` (`ironic.api.controllers.v1.node.NodeManagementController` attribute), 856
- `boot_index` (`ironic.db.sqlalchemy.models.VolumeTarget` attribute), 1780
- `boot_index` (`ironic.objects.volume_target.VolumeTarget` property), 3386
- `boot_index` (`ironic.objects.volume_target.VolumeTargetCRUDPayload` property), 3405
- `boot_instance()` (`ironic.drivers.modules.agent_base.AgentOobStepsMixin` method), 2575
- `boot_interface` (`ironic.db.sqlalchemy.models.Node` attribute), 1772
- `boot_interface` (`ironic.objects.node.Node` property), 3262
- `boot_interface` (`ironic.objects.node.NodeCorrectedPowerStatePayload` property), 3291
- `boot_interface` (`ironic.objects.node.NodeCRUDPayload` property), 3283
- `boot_interface` (`ironic.objects.node.NodePayload` property), 3296
- `boot_interface` (`ironic.objects.node.NodeSetPowerStatePayload` property), 3302
- `boot_interface` (`ironic.objects.node.NodeSetProvisionStatePayload` property), 3307

- (*ironic.objects.node.NodeSetProvisionStatePayload* method), 2107
 property), 3307
- `boot_into_iso()` (*ironic.drivers.modules.ilo.vendor.VendorPassthru* method), 2262
 method), 2255
- `boot_up_seq()` (*ironic.drivers.modules.ibm.vendor.IBMCVMe* method), 2099
 method), 2867
- `BootDeviceController` (class in *ironic.api.controllers.v1.node*), 843
- `BootInterface` (class in *ironic.drivers.base*), 2976
- `build_agent_options()` (in module *ironic.drivers.modules.deploy_utils*), 2651
- `build_configdrive()` (in module *ironic.conductor.utils*), 1575
- `build_deploy_pxe_options()` (in module *ironic.common.pxe_utils*), 1269
- `build_driver_for_task()` (in module *ironic.common.driver_factory*), 1138
- `build_extra_pxe_options()` (in module *ironic.common.pxe_utils*), 1269
- `build_instance_info_for_deploy()` (in module *ironic.drivers.modules.deploy_utils*), 2651
- `build_instance_pxe_options()` (in module *ironic.common.pxe_utils*), 1269
- `build_kickstart_config_options()` (in module *ironic.common.pxe_utils*), 1269
- `build_pxe_config_options()` (in module *ironic.common.pxe_utils*), 1270
- `build_service_pxe_config()` (in module *ironic.common.pxe_utils*), 1274
- `build_url()` (in module *ironic.api.controllers.link*), 1056
- ## C
- `cache_bios_settings()` (in module *ironic.drivers.base*), 3096
- `cache_bios_settings()` (*ironic.drivers.base.BIOSInterface* method), 2954
- `cache_bios_settings()` (*ironic.drivers.modules.drac.bios.DracWSMBoot* method), 1946
- `cache_bios_settings()` (*ironic.drivers.modules.fake.FakeBIOS* method), 2699
- `cache_bios_settings()` (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 2107
 property), 3307
- `cache_bios_settings()` (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 2262
 method), 2255
- `cache_bios_settings()` (*ironic.drivers.modules.noop.NoBIOS* method), 2867
- `cache_bios_settings()` (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 2380
- `cache_instance_image()` (in module *ironic.drivers.modules.deploy_utils*), 2653
- `cache_ramdisk_kernel()` (in module *ironic.common.pxe_utils*), 1274
- `call()` (*ironic.common.glance_service.image_service.GlanceImageService* method), 1084
- `can_send_create_port()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1432
- `can_send_rescue()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1432
- `candidate_nodes` (*ironic.db.sqlalchemy.models.Allocation* attribute), 1765
- `candidate_nodes` (*ironic.objects.allocation.Allocation* property), 3154
- `candidate_nodes` (*ironic.objects.allocation.AllocationCRUDPayload* property), 3167
- `capabilities` (*ironic.drivers.base.BootInterface* attribute), 2976
- `capabilities` (*ironic.drivers.modules.fake.FakeBoot* attribute), 2705
- `capabilities` (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* attribute), 2117
- `capabilities` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* attribute), 2128
- `capabilities` (*ironic.drivers.modules.ipxe.iPXEBoot* attribute), 2862
- `capabilities` (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* attribute), 2270
- `capabilities` (*ironic.drivers.modules.pxe.PXEBoot* attribute), 2908
- `capabilities` (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* attribute), 2392
- `capabilities_to_dict()` (in module *ironic.drivers.utils*), 3141
- `CatalogNotFound`, 1154

- CDROM (in module *ironic.common.boot_devices*), 1112
- `change_node_power_state()` (*ironic.conductor.manager.ConductorManager* method), 1396
- `change_node_power_state()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1432
- `change_physical_disk_state()` (in module *ironic.drivers.modules.drac.raid*), 2031
- Chassis (class in *ironic.db.sqlalchemy.models*), 1767
- Chassis (class in *ironic.objects.chassis*), 3197
- CHASSIS (in module *ironic.common.components*), 1129
- `chassis_id` (*ironic.db.sqlalchemy.models.Node* attribute), 1772
- `chassis_id` (*ironic.objects.node.Node* property), 3262
- `chassis_uuid` (*ironic.objects.node.NodeCRUDPayload* property), 3283
- ChassisAlreadyExists, 1154
- ChassisController (class in *ironic.api.controllers.v1.chassis*), 813
- ChassisCRUDNotification (class in *ironic.objects.chassis*), 3206
- ChassisCRUDPayload (class in *ironic.objects.chassis*), 3207
- ChassisNotEmpty, 1154
- ChassisNotFound, 1154
- `check()` (in module *ironic.common.policy*), 1262
- `check_allocation_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 956
- `check_allow_clean_disable_ramdisk()` (in module *ironic.api.controllers.v1.utils*), 958
- `check_allow_configdrive()` (in module *ironic.api.controllers.v1.utils*), 958
- `check_allow_deploy_steps()` (in module *ironic.api.controllers.v1.utils*), 958
- `check_allow_driver_detail()` (in module *ironic.api.controllers.v1.utils*), 958
- `check_allow_filter_by_conductor()` (in module *ironic.api.controllers.v1.utils*), 959
- `check_allow_filter_by_conductor_group()` (in module *ironic.api.controllers.v1.utils*), 959
- `check_allow_filter_by_fault()` (in module *ironic.api.controllers.v1.utils*), 960
- `check_allow_filter_by_lessee()` (in module *ironic.api.controllers.v1.utils*), 961
- `check_allow_filter_by_owner()` (in module *ironic.api.controllers.v1.utils*), 961
- `check_allow_filter_driver_type()` (in module *ironic.api.controllers.v1.utils*), 962
- `check_allow_management_verbs()` (in module *ironic.api.controllers.v1.utils*), 962
- `check_allow_specify_driver()` (in module *ironic.api.controllers.v1.utils*), 962
- `check_allow_specify_fields()` (in module *ironic.api.controllers.v1.utils*), 963
- `check_allow_specify_resource_class()` (in module *ironic.api.controllers.v1.utils*), 964
- `check_allowed_fields()` (in module *ironic.api.controllers.v1.utils*), 964
- `check_allowed_portgroup_fields()` (in module *ironic.api.controllers.v1.utils*), 965
- `check_and_update_node_interfaces()` (in module *ironic.common.driver_factory*), 1140
- `check_cipher_suite_errors()` (in module *ironic.drivers.modules.ipmitool*), 2854
- `check_dir()` (in module *ironic.common.utils*), 1342
- `check_for_invalid_fields()` (in module *ironic.api.controllers.v1.utils*), 966
- `check_for_invalid_state_and_allow_filter()` (in module *ironic.api.controllers.v1.utils*), 967
- `check_for_missing_params()` (in module *ironic.drivers.modules.deploy_utils*), 2654
- `check_image_service()` (in module *ironic.common.glance_service.image_service*), 1091
- `check_image_size()` (in module *ironic.drivers.modules.agent*), 2550
- `check_interface_capability()` (in module *ironic.drivers.modules.deploy_utils*), 2656
- `check_list_policy()` (in module *ironic.api.controllers.v1.utils*), 967
- `check_multiple_node_policies_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 969
- `check_node_list()` (*ironic.db.api.Connection* method), 1785

- `check_node_list()` (*ironic.db.sqlalchemy.api.Connection* method), 1631
- `check_node_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 970
- `check_obj_versions()` (*ironic.cmd.dbsync.DBCommand* method), 1082
- `check_owner_policy()` (in module *ironic.api.controllers.v1.utils*), 972
- `check_policy()` (in module *ironic.api.controllers.v1.utils*), 974
- `check_policy()` (in module *ironic.common.policy*), 1263
- `check_policy_true()` (in module *ironic.api.controllers.v1.utils*), 974
- `check_port_list_policy()` (in module *ironic.api.controllers.v1.utils*), 975
- `check_port_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 977
- `check_share_fs_mounted()` (in module *ironic.drivers.modules.irmc.boot*), 2282
- `check_versions()` (*ironic.db.api.Connection* method), 1786
- `check_versions()` (*ironic.db.sqlalchemy.api.Connection* method), 1632
- `check_volume_list_policy()` (in module *ironic.api.controllers.v1.utils*), 979
- `check_volume_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 980
- Checks (class in *ironic.cmd.status*), 1083
- `choose_cipher_suite()` (in module *ironic.drivers.modules.ipmitool*), 2855
- CinderStorage (class in *ironic.drivers.modules.storage.cinder*), 2490
- `clean_dhcp()` (*ironic.common.dhcp_factory.DHCPFactory* method), 1132
- `clean_dhcp_opts()` (*ironic.dhcp.base.BaseDHCP* method), 1912
- CLEAN_FAILURE (in module *ironic.common.faults*), 1176
- `clean_step` (*ironic.db.sqlalchemy.models.Node* attribute), 1772
- `clean_step` (*ironic.objects.node.Node* property), 3262
- `clean_step` (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3291
- `clean_step` (*ironic.objects.node.NodeCRUDPayload* property), 3283
- `clean_step` (*ironic.objects.node.NodePayload* property), 3296
- `clean_step` (*ironic.objects.node.NodeSetPowerStatePayload* property), 3302
- `clean_step` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3307
- `clean_step()` (in module *ironic.drivers.base*), 3096
- `clean_up()` (*ironic.drivers.base.DeployInterface* method), 2987
- `clean_up()` (*ironic.drivers.base.RescueInterface* method), 3084
- `clean_up()` (*ironic.drivers.modules.agent.AgentRescue* method), 2534
- `clean_up()` (*ironic.drivers.modules.agent.CustomAgentDeploy* method), 2542
- `clean_up()` (*ironic.drivers.modules.agent_base.AgentBaseMixin* method), 2553
- `clean_up()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 1934
- `clean_up()` (*ironic.drivers.modules.fake.FakeDeploy* method), 2719
- `clean_up()` (*ironic.drivers.modules.image_cache.ImageCache* method), 2788
- `clean_up_caches()` (in module *ironic.drivers.modules.image_cache*), 2791
- `clean_up_instance()` (*ironic.drivers.base.BootInterface* method), 2976
- `clean_up_instance()` (*ironic.drivers.modules.fake.FakeBoot* method), 2705
- `clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloIPXEBoot* method), 2140
- `clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloPXEBot* method), 2112
- `clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* method), 2117
- `clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 2129
- `clean_up_instance()` (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoo*

method), 2270

clean_up_instance() (in module *ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 2914

clean_up_instance() (in module *ironic.drivers.modules.redfish.boot.RedfishBootMixin* method), 2392

clean_up_pxe_config() (in module *ironic.common.pxe_utils*), 1274

clean_up_pxe_env() (in module *ironic.common.pxe_utils*), 1274

clean_up_ramdisk() (in module *ironic.drivers.base.BootInterface* method), 2977

clean_up_ramdisk() (in module *ironic.drivers.modules.fake.FakeBoot* method), 2706

clean_up_ramdisk() (in module *ironic.drivers.modules.ilo.boot.IloUefiHttpBoot* method), 2118

clean_up_ramdisk() (in module *ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 2130

clean_up_ramdisk() (in module *ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* method), 2271

clean_up_ramdisk() (in module *ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 2915

clean_up_ramdisk() (in module *ironic.drivers.modules.redfish.boot.RedfishBootMixin* method), 2392

CLEANFAIL (in module *ironic.common.states*), 1318

CLEANING (in module *ironic.common.states*), 1318

cleaning_error_handler() (in module *ironic.conductor.utils*), 1577

cleanup() (in module *ironic.common.rpc*), 1313

cleanup() (in module *ironic.drivers.modules.image_cache*), 2793

cleanup_after_timeout() (in module *ironic.conductor.utils*), 1579

cleanup_cleanwait_timeout() (in module *ironic.conductor.utils*), 1579

cleanup_disk_image() (in module *ironic.drivers.modules.image_utils*), 2796

cleanup_floppy_image() (in module *ironic.drivers.modules.image_utils*), 2797

cleanup_iso_image() (in module *ironic.drivers.modules.image_utils*), 2798

cleanup_rescuewait_timeout() (in module *ironic.conductor.utils*), 1580

clean_media_boot() (in module *ironic.drivers.modules.ilo.common*), 2155

CLEANWAIT (in module *ironic.common.states*), 1319

clear_certificates() (in module *ironic.drivers.modules.ilo.common*), 2156

clear_foreign_config() (in module *ironic.drivers.modules.drac.raid*), 2033

clear_iscsi_boot_target() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 2208

clear_job_queue() (in module *ironic.drivers.modules.drac.management.DracRedfishManagement* method), 1987

clear_job_queue() (in module *ironic.drivers.modules.drac.management.DracWSManagement* method), 1995

clear_node_reservations_for_conductor() (in module *ironic.db.sqlalchemy.api.Connection* method), 1634

clear_node_target_power_state() (in module *ironic.db.sqlalchemy.api.Connection* method), 1634

clear_secure_boot_keys() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 2209

clear_secure_boot_keys() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 2408

ClientSideError, 1154

code (in module *ironic.common.exception.Conflict* attribute), 1155

code (in module *ironic.common.exception.Invalid* attribute), 1163

code (in module *ironic.common.exception.NoFreeConductorWorker* attribute), 1167

code (in module *ironic.common.exception.NotAcceptable* attribute), 1169

code (in module *ironic.common.exception.NotAuthorized* attribute), 1169

code (in module *ironic.common.exception.NotFound* attribute), 1169

code (in module *ironic.common.exception.TemporaryFailure* attribute), 1173

[coerce\(\)](#) (*ironic.objects.fields.FlexibleDict* static method), 3242
[coerce\(\)](#) (*ironic.objects.fields.MACAddress* static method), 3245
[coerce\(\)](#) (*ironic.objects.fields.StringAcceptsCallable* static method), 3248
[collect_ramdisk_logs\(\)](#) (in module *ironic.drivers.utils*), 3142
[collect_system_logs\(\)](#) (*ironic.drivers.modules.agent_client.AgentClient* method), 2590
[collection_from_list\(\)](#) (in module *ironic.api.controllers.v1.bios*), 812
[commit_bios_config\(\)](#) (*ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthru* method), 2054
[commit_config\(\)](#) (in module *ironic.drivers.modules.drac.bios*), 1952
[commit_config\(\)](#) (in module *ironic.drivers.modules.drac.raid*), 2035
[CommunicationError](#), 1155
[compute_image_checksum\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 2657
[Conductor](#) (class in *ironic.db.sqlalchemy.models*), 1768
[Conductor](#) (class in *ironic.objects.conductor*), 3208
[conductor_affinity](#) (*ironic.db.sqlalchemy.models.Allocation* attribute), 1765
[conductor_affinity](#) (*ironic.db.sqlalchemy.models.Node* attribute), 1772
[conductor_affinity](#) (*ironic.objects.allocation.Allocation* property), 3154
[conductor_affinity](#) (*ironic.objects.node.Node* property), 3263
[conductor_group](#) (*ironic.db.sqlalchemy.models.Conductor* attribute), 1768
[conductor_group](#) (*ironic.db.sqlalchemy.models.Node* attribute), 1772
[conductor_group](#) (*ironic.objects.conductor.Conductor* property), 3208
[conductor_group](#) (*ironic.objects.node.Node* property), 3263
[conductor_group](#)
[conductor_group](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3291
[conductor_group](#) (*ironic.objects.node.NodeCRUDPayload* property), 3283
[conductor_group](#) (*ironic.objects.node.NodePayload* property), 3296
[conductor_group](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 3302
[conductor_group](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3307
[ConductorHardware](#) (*ironic.db.sqlalchemy.models.ConductorHardware* attribute), 1769
[ConductorAlreadyRegistered](#), 1155
[ConductorAPI](#) (class in *ironic.conductor.rpcapi*), 1418
[ConductorHardwareInterfaces](#) (class in *ironic.db.sqlalchemy.models*), 1769
[ConductorHardwareInterfacesAlreadyRegistered](#), 1155
[ConductorManager](#) (class in *ironic.conductor.manager*), 1395
[ConductorNotFound](#), 1155
[ConductorsController](#) (class in *ironic.api.controllers.v1.conductor*), 822
[config\(\)](#) (in module *ironic.api.controllers.v1.ramdisk*), 930
[ConfigHook](#) (class in *ironic.api.hooks*), 1071
[ConfigInvalid](#), 1155
[ConfigNotFound](#), 1155
[configure_intel_speedselect\(\)](#) (*ironic.drivers.modules.intel_ipmi.management.IntelIPMI* method), 2260
[configure_local_boot\(\)](#) (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2561
[configure_secure_boot_if_needed\(\)](#) (in module *ironic.drivers.modules.boot_mode_utils*), 2634
[configure_tenant_networks\(\)](#) (*ironic.drivers.base.NetworkInterface* method), 3046
[configure_tenant_networks\(\)](#) (*ironic.drivers.modules.network.flat.FlatNetwork* method), 2353

configure_tenant_networks() (*ironic.drivers.modules.network.neutron.NeutronNetworkProperty*), 3283
 (*ironic.drivers.modules.network.noop.NoopNetworkProperty*), 3297
 method), 2362 console_interface
 configure_tenant_networks() (*ironic.objects.node.NodePayloadProperty*), 3296
 method), 2370 console_interface
 Conflict, 1155 (*ironic.objects.node.NodeSetPowerStatePayloadProperty*), 3302
 Connection (*class in ironic.db.api*), 1782
 Connection (*class in ironic.db.sqlalchemy.api*), 1628 console_interface
 (*ironic.objects.node.NodeSetProvisionStatePayloadProperty*), 3308
 connection (*ironic.common.swift.SwiftAPIAttribute*), 1334 ConsoleError, 1155
 connector_id (*ironic.db.sqlalchemy.models.VolumeConnectorAttribute*), 1779 ConsoleInterface (*class in ironic.drivers.base*),
 2983
 connector_id (*ironic.objects.volume_connector.VolumeConnectorProperty*), 3367 ConsoleSubprocessFailed, 1156
 connector_id (*ironic.objects.volume_connector.VolumeConnectorProperty*), 3385 ContextHook (*class in ironic.api.hooks*), 1072
 console (*ironic.api.controllers.v1.node.NodeStatesControllerMethod*), 2576
 attribute), 857 console (*ironic.drivers.modules.agent_base.HeartbeatMixin*)
 console (*ironic.drivers.base.BareDriverAttribute*), 2959 continue_node_clean() (*in module*
 console_enabled (*ironic.db.sqlalchemy.models.NodeAttribute*), 1772 *ironic.conductor.cleaning*), 1383
 console_enabled (*ironic.objects.node.NodeProperty*), 3263 continue_node_clean() (*ironic.conductor.manager.ConductorManager*
 console_enabled (*ironic.objects.node.NodeCorrectedPowerStatePayloadProperty*), 3291 method), 1396
 console_enabled (*ironic.objects.node.NodeCRUDPayloadProperty*), 3283 continue_node_clean() (*ironic.conductor.rpcapi.ConductorAPI*
 console_enabled (*ironic.objects.node.NodePayloadProperty*), 3296 method), 1434
 console_enabled (*ironic.objects.node.NodeSetPowerStatePayloadProperty*), 3302 continue_node_deploy() (*in module*
 console_enabled (*ironic.objects.node.NodeSetProvisionStatePayloadProperty*), 3307 *ironic.conductor.deployments*), 1388
 console_interface (*ironic.db.sqlalchemy.models.NodeAttribute*), 1772 continue_node_deploy() (*ironic.conductor.manager.ConductorManager*
 console_interface (*ironic.objects.node.NodeProperty*), 3263 method), 1398
 console_interface (*ironic.objects.node.NodeCorrectedPowerStatePayloadProperty*), 3291 continue_node_deploy() (*ironic.conductor.rpcapi.ConductorAPI*
 console_interface (*ironic.objects.node.NodeCRUDPayloadProperty*), 3283 Controller (*class in ironic.api.controllers.v1*),
 console_interface (*ironic.objects.node.NodePayloadProperty*), 3296 1053
 console_interface (*ironic.objects.node.NodeSetPowerStatePayloadProperty*), 3302 convert() (*in module*
 console_interface (*ironic.objects.node.NodeSetProvisionStatePayloadProperty*), 3307 *ironic.api.controllers.v1.volume*), 1023
 console_interface convert_drive_units() (*in module*
 (*ironic.db.sqlalchemy.models.NodeAttribute*), 1772 *ironic.drivers.modules.redfish.raid*),
 console_interface convert_steps() (*in module*
 (*ironic.objects.node.NodeProperty*), 3263 *ironic.api.controllers.v1.deploy_template*),
 console_interface convert_to_version() 831
 (*ironic.objects.node.NodeCorrectedPowerStatePayloadProperty*), 3291 (*ironic.objects.base.IronicObject*
 console_interface convert_with_links() (*in module*
 (*ironic.objects.node.NodeCRUDPayloadProperty*), 3283 *ironic.api.controllers.v1.allocation*),
 console_interface 811
 (*ironic.objects.node.NodeSetProvisionStatePayloadProperty*), 3307

- `convert_with_links()` (in module `ironic.api.controllers.v1.bios`), 812
- `convert_with_links()` (in module `ironic.api.controllers.v1.chassis`), 819
- `convert_with_links()` (in module `ironic.api.controllers.v1.conductor`), 825
- `convert_with_links()` (in module `ironic.api.controllers.v1.deploy_template`), 831
- `convert_with_links()` (in module `ironic.api.controllers.v1.driver`), 839
- `convert_with_links()` (in module `ironic.api.controllers.v1.port`), 915
- `convert_with_links()` (in module `ironic.api.controllers.v1.portgroup`), 923
- `convert_with_links()` (in module `ironic.api.controllers.v1.ramdisk`), 930
- `convert_with_links()` (in module `ironic.api.controllers.v1.volume_connector`), 1038
- `convert_with_links()` (in module `ironic.api.controllers.v1.volume_target`), 1053
- `converted_size()` (in module `ironic.common.images`), 1197
- `copy_image_to_swift()` (in module `ironic.drivers.modules.ilo.common`), 2158
- `copy_image_to_web_server()` (in module `ironic.drivers.modules.ilo.common`), 2159
- `core_interfaces` (`ironic.drivers.base.BareDriver` property), 2959
- `create()` (`ironic.objects.allocation.Allocation` method), 3154
- `create()` (`ironic.objects.bios.BIOSSetting` method), 3182
- `create()` (`ironic.objects.bios.BIOSSettingList` class method), 3189
- `create()` (`ironic.objects.chassis.Chassis` method), 3197
- `create()` (`ironic.objects.deploy_template.DeployTemplate` method), 3217
- `create()` (`ironic.objects.deployment.Deployment` method), 3233
- `create()` (`ironic.objects.node.Node` method), 3263
- `create()` (`ironic.objects.port.Port` method), 3318
- `create()` (`ironic.objects.portgroup.Portgroup` method), 3340
- `create()` (`ironic.objects.trait.Trait` method), 3357
- `create()` (`ironic.objects.trait.TraitList` class method), 3362
- `create()` (`ironic.objects.volume_connector.VolumeConnector` method), 3367
- `create()` (`ironic.objects.volume_target.VolumeTarget` method), 3386
- `create_allocation()` (`ironic.conductor.manager.ConductorManager` method), 1400
- `create_allocation()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1437
- `create_allocation()` (`ironic.db.api.Connection` method), 1787
- `create_allocation()` (`ironic.db.sqlalchemy.api.Connection` method), 1634
- `create_bios_setting_list()` (`ironic.db.api.Connection` method), 1788
- `create_bios_setting_list()` (`ironic.db.sqlalchemy.api.Connection` method), 1635
- `create_boot_iso()` (in module `ironic.common.images`), 1198
- `create_chassis()` (`ironic.db.api.Connection` method), 1791
- `create_chassis()` (`ironic.db.sqlalchemy.api.Connection` method), 1638
- `create_configuration()` (`ironic.drivers.base.RAIDInterface` method), 3074
- `create_configuration()` (`ironic.drivers.modules.agent.AgentRAID` method), 2528
- `create_configuration()` (`ironic.drivers.modules.drac.raid.DracRedfishRAID` method), 2018
- `create_configuration()` (`ironic.drivers.modules.drac.raid.DracWSManRAID` method), 2025
- `create_configuration()` (`ironic.drivers.modules.fake.FakeRAID` method), 2766
- `create_configuration()`

(ironic.drivers.modules.ibmraid.IbmRAID method), 1643
method), 2093
create_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID method*), 2250
create_configuration() (*ironic.drivers.modules.irmc.raid.IRMCRaid method*), 2326
create_configuration() (*ironic.drivers.modules.noop.NoRAID method*), 2876
create_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID method*), 2460
create_deploy_template() (*ironic.db.api.Connection method*), 1792
create_deploy_template() (*ironic.db.sqlalchemy.api.Connection method*), 1638
create_esp_image_for_uefi() (*in module ironic.common.images*), 1203
create_ipxe_boot_script() (*in module ironic.common.pxe_utils*), 1276
create_isolinux_image_for_bios() (*in module ironic.common.images*), 1206
create_link_without_raise() (*in module ironic.common.utils*), 1344
create_node() (*ironic.conductor.manager.ConductorManager method*), 1400
create_node() (*ironic.conductor.rpcapi.ConductorAPI method*), 1437
create_node() (*ironic.db.api.Connection method*), 1793
create_node() (*ironic.db.sqlalchemy.api.Connection method*), 1640
create_object() (*ironic.common.swift.SwiftAPI method*), 1335
create_port() (*ironic.conductor.manager.ConductorManager method*), 1400
create_port() (*ironic.conductor.rpcapi.ConductorAPI method*), 1440
create_port() (*ironic.db.api.Connection method*), 1796
create_port() (*ironic.db.sqlalchemy.api.Connection method*), 1642
create_portgroup() (*ironic.db.api.Connection method*), 1796
create_portgroup() (*ironic.db.sqlalchemy.api.Connection method*), 1643
create_ports_if_not_exist() (*in module ironic.drivers.modules.inspect_utils*), 2811
create_pxe_config() (*in module ironic.common.pxe_utils*), 1276
create_schema command line option --help, 538 -h, 538
create_schema() (*in module ironic.db.migration*), 1911
create_schema() (*in module ironic.db.sqlalchemy.migration*), 1762
create_schema() (*ironic.cmd.dbsync.DBCommand method*), 1083
create_schema, ironic-dbsync command line option, 537
create_vfat_image() (*in module ironic.common.images*), 1209
create_virtual_disk() (*in module ironic.drivers.modules.drac.raid*), 2037
create_virtual_disk() (*in module ironic.drivers.modules.redfish.raid*), 2474
create_volume_connector() (*ironic.db.api.Connection method*), 1797
create_volume_connector() (*ironic.db.sqlalchemy.api.Connection method*), 1644
create_volume_target() (*ironic.db.api.Connection method*), 1800
create_volume_target() (*ironic.db.sqlalchemy.api.Connection method*), 1646
created_at (*ironic.db.sqlalchemy.models.Allocation attribute*), 1766
created_at (*ironic.db.sqlalchemy.models.BIOSSetting attribute*), 1767
created_at (*ironic.db.sqlalchemy.models.Chassis attribute*), 1768
created_at (*ironic.db.sqlalchemy.models.Conductor attribute*), 1768
created_at (*ironic.db.sqlalchemy.models.ConductorHardwareInt attribute*), 1769
created_at (*ironic.db.sqlalchemy.models.DeployTemplate attribute*), 1770
created_at (*ironic.db.sqlalchemy.models.DeployTemplateStep attribute*), 1770

attribute), 1771
 created_at (*ironic.db.sqlalchemy.models.Node*
 attribute), 1773
 created_at (*ironic.db.sqlalchemy.models.NodeTag*
 attribute), 1775
 created_at (*ironic.db.sqlalchemy.models.NodeTrait*
 attribute), 1776
 created_at (*ironic.db.sqlalchemy.models.Port*
 attribute), 1777
 created_at (*ironic.db.sqlalchemy.models.Portgroup*
 attribute), 1778
 created_at (*ironic.db.sqlalchemy.models.VolumeConnector*
 property), 1779
 created_at (*ironic.db.sqlalchemy.models.VolumeTarget*
 attribute), 1780
 created_at (*ironic.objects.allocation.Allocation*
 property), 3155
 created_at (*ironic.objects.allocation.AllocationCRUDNotification*
 property), 3166
 created_at (*ironic.objects.allocation.AllocationCRUDPayload*
 property), 3167
 created_at (*ironic.objects.bios.BIOSSetting*
 property), 3184
 created_at (*ironic.objects.bios.BIOSSettingList*
 property), 3191
 created_at (*ironic.objects.chassis.Chassis* *prop-*
 erty), 3198
 created_at (*ironic.objects.chassis.ChassisCRUDNotification*
 property), 3207
 created_at (*ironic.objects.chassis.ChassisCRUDPayload*
 property), 3207
 created_at (*ironic.objects.conductor.Conductor*
 property), 3208
 created_at (*ironic.objects.deploy_template.DeployTemplate*
 property), 3219
 created_at (*ironic.objects.deploy_template.DeployTemplateCRUDNotification*
 property), 3231
 created_at (*ironic.objects.deploy_template.DeployTemplateCRUDPayload*
 property), 3232
 created_at (*ironic.objects.deployment.Deployment*
 property), 3235
 created_at (*ironic.objects.node.Node* *property*),
 3264
 created_at (*ironic.objects.node.NodeConsoleNotification*
 property), 3288
 created_at (*ironic.objects.node.NodeCorrectedPowerStateNotification*
 property), 3289
 created_at (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 property), 3291
 created_at (*ironic.objects.node.NodeCRUDNotification*
 property), 3281
 created_at (*ironic.objects.node.NodeCRUDPayload*
 property), 3283
 created_at (*ironic.objects.node.NodeMaintenanceNotification*
 property), 3295
 created_at (*ironic.objects.node.NodePayload*
 property), 3297
 created_at (*ironic.objects.node.NodeSetPowerStateNotification*
 property), 3301
 created_at (*ironic.objects.node.NodeSetPowerStatePayload*
 property), 3302
 created_at (*ironic.objects.node.NodeSetProvisionStateNotification*
 property), 3306
 created_at (*ironic.objects.node.NodeSetProvisionStatePayload*
 property), 3308
 created_at (*ironic.objects.notification.EventType*
 property), 3312
 created_at (*ironic.objects.notification.NotificationBase*
 property), 3314
 created_at (*ironic.objects.notification.NotificationPayloadBase*
 property), 3315
 created_at (*ironic.objects.notification.NotificationPublisher*
 property), 3317
 created_at (*ironic.objects.port.Port* *property*),
 3319
 created_at (*ironic.objects.port.PortCRUDNotification*
 property), 3337
 created_at (*ironic.objects.port.PortCRUDPayload*
 property), 3338
 created_at (*ironic.objects.portgroup.Portgroup*
 property), 3341
 created_at (*ironic.objects.portgroup.PortgroupCRUDNotification*
 property), 3355
 created_at (*ironic.objects.portgroup.PortgroupCRUDPayload*
 property), 3356
 created_at (*ironic.objects.trait.Trait* *property*),
 3364
 created_at (*ironic.objects.trait.TraitList* *prop-*
 erty), 3369
 created_at (*ironic.objects.volume_connector.VolumeConnector*
 property), 3384
 created_at (*ironic.objects.volume_connector.VolumeConnectorC*
 onsoleNotification *property*), 3385
 created_at (*ironic.objects.volume_target.VolumeTarget*
 property), 3387
 created_at (*ironic.objects.volume_target.VolumeTargetCRUDNoti-*
 fication *property*), 3404
 created_at (*ironic.objects.volume_target.VolumeTargetCRUDPay-*
 load *property*), 3405
 CRITICAL (*ironic.objects.fields.NotificationLevel*

attribute), 3246
 CustomAgentDeploy (class in *ironic.drivers.modules.agent*), 2541

D

DatabaseVersionTooOld, 1156
 datatype (*ironic.api.functions.FunctionArgument* attribute), 1065
 DateTimeField (class in *ironic.objects.fields*), 3242
 dbapi (*ironic.objects.allocation.Allocation* attribute), 3155
 dbapi (*ironic.objects.bios.BIOSSetting* attribute), 3184
 dbapi (*ironic.objects.bios.BIOSSettingList* attribute), 3191
 dbapi (*ironic.objects.chassis.Chassis* attribute), 3198
 dbapi (*ironic.objects.conductor.Conductor* attribute), 3208
 dbapi (*ironic.objects.deploy_template.DeployTemplate* attribute), 3219
 dbapi (*ironic.objects.deployment.Deployment* attribute), 3235
 dbapi (*ironic.objects.node.Node* attribute), 3264
 dbapi (*ironic.objects.port.Port* attribute), 3319
 dbapi (*ironic.objects.portgroup.Portgroup* attribute), 3341
 dbapi (*ironic.objects.trait.Trait* attribute), 3358
 dbapi (*ironic.objects.trait.TraitList* attribute), 3364
 dbapi (*ironic.objects.volume_connector.VolumeConnector* attribute), 3369
 dbapi (*ironic.objects.volume_target.VolumeTarget* attribute), 3387
 DBCommand (class in *ironic.cmd.dbsync*), 1081
 DBHook (class in *ironic.api.hooks*), 1074
 DEBUG (*ironic.objects.fields.NotificationLevel* attribute), 3246
 deconfigure_secure_boot_if_needed() (in module *ironic.drivers.modules.boot_mode_utils*), 2634
 default (*ironic.api.functions.FunctionArgument* attribute), 1065
 default (*ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 1769
 default_interface() (in module *ironic.common.driver_factory*), 1143
 default_version() (in module *ironic.api.controllers.version*), 1057

del_host() (*ironic.conductor.base_manager.BaseConductorManager* method), 1378
 delete() (*ironic.api.controllers.v1.allocation.AllocationsController* method), 805
 delete() (*ironic.api.controllers.v1.allocation.NodeAllocationController* method), 811
 delete() (*ironic.api.controllers.v1.chassis.ChassisController* method), 813
 delete() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 826
 delete() (*ironic.api.controllers.v1.node.NodeMaintenanceController* method), 855
 delete() (*ironic.api.controllers.v1.node.NodesController* method), 878
 delete() (*ironic.api.controllers.v1.node.NodeTraitsController* method), 872
 delete() (*ironic.api.controllers.v1.node.NodeVIFController* method), 874
 delete() (*ironic.api.controllers.v1.port.PortsController* method), 905
 delete() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 916
 delete() (*ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 1024
 delete() (*ironic.api.controllers.v1.volume_target.VolumeTargetController* method), 1039
 delete() (*ironic.objects.bios.BIOSSetting* class method), 3184
 delete() (*ironic.objects.bios.BIOSSettingList* class method), 3191
 DELETE_ALLOWED_STATES (in module *ironic.common.states*), 1320
 delete_bios_setting_list() (*ironic.db.api.Connection* method), 1802
 delete_bios_setting_list() (*ironic.db.sqlalchemy.api.Connection* method), 1649
 delete_configuration() (*ironic.drivers.base.RAIDInterface* method), 3076
 delete_configuration() (*ironic.drivers.modules.agent.AgentRAID* method), 2530
 delete_configuration() (*ironic.drivers.modules.drac.raid.DracRedfishRAID* method), 2021
 delete_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID* method), 2028
 delete_configuration()

- [\(ironic.drivers.modules.fake.FakeRAID method\), 2769](#)
- [delete_configuration\(\) \(ironic.drivers.modules.ibmcr.raid.IbmcrRAID method\), 2095](#)
- [delete_configuration\(\) \(ironic.drivers.modules.ilo.raid.Ilo5RAID method\), 2253](#)
- [delete_configuration\(\) \(ironic.drivers.modules.irmc.raid.IRMCRRAID method\), 2328](#)
- [delete_configuration\(\) \(ironic.drivers.modules.noop.NoRAID method\), 2878](#)
- [delete_configuration\(\) \(ironic.drivers.modules.redfish.raid.RedfishRAID method\), 2463](#)
- [delete_node_tag\(\) \(ironic.db.api.Connection method\), 1804](#)
- [delete_node_tag\(\) \(ironic.db.sqlalchemy.api.Connection method\), 1650](#)
- [delete_node_trait\(\) \(ironic.db.api.Connection method\), 1804](#)
- [delete_node_trait\(\) \(ironic.db.sqlalchemy.api.Connection method\), 1651](#)
- [delete_object\(\) \(ironic.common.swift.SwiftAPI method\), 1336](#)
- [delete_virtual_disk\(\) \(in module ironic.drivers.modules.drac.raid\), 2040](#)
- [DELETED \(in module ironic.common.states\), 1319](#)
- [DELETING \(in module ironic.common.states\), 1320](#)
- [DEPLOY \(in module ironic.common.states\), 1320](#)
- [deploy \(ironic.drivers.base.BareDriver attribute\), 2959](#)
- [deploy\(\) \(ironic.drivers.base.DeployInterface method\), 2988](#)
- [deploy\(\) \(ironic.drivers.modules.agent.CustomAgentDeploy method\), 2543](#)
- [deploy\(\) \(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method\), 1934](#)
- [deploy\(\) \(ironic.drivers.modules.fake.FakeDeploy method\), 2720](#)
- [deploy\(\) \(ironic.drivers.modules.pxe.PXEAnacondaDeploy method\), 2902](#)
- [deploy\(\) \(ironic.drivers.modules.pxe.PXERamdiskDeploy method\), 2908](#)
- [deploy_has_started\(\)](#)
- [\(ironic.drivers.modules.pxe.PXEAnacondaDeploy method\), 2903](#)
- [deploy_interface \(ironic.db.sqlalchemy.models.Node attribute\), 1773](#)
- [deploy_interface \(ironic.objects.node.Node property\), 3264](#)
- [deploy_interface \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 3291](#)
- [deploy_interface \(ironic.objects.node.NodeCRUDPayload property\), 3284](#)
- [deploy_interface \(ironic.objects.node.NodePayload property\), 3297](#)
- [deploy_interface \(ironic.objects.node.NodeSetPowerStatePayload property\), 3302](#)
- [deploy_interface \(ironic.objects.node.NodeSetProvisionStatePayload property\), 3308](#)
- [deploy_is_done\(\) \(ironic.drivers.modules.pxe.PXEAnacondaDeploy method\), 2903](#)
- [deploy_step \(ironic.db.sqlalchemy.models.Node attribute\), 1773](#)
- [deploy_step \(ironic.objects.node.Node property\), 3264](#)
- [deploy_step \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 3291](#)
- [deploy_step \(ironic.objects.node.NodeCRUDPayload property\), 3284](#)
- [deploy_step \(ironic.objects.node.NodePayload property\), 3297](#)
- [deploy_step \(ironic.objects.node.NodeSetPowerStatePayload property\), 3302](#)
- [deploy_step \(ironic.objects.node.NodeSetProvisionStatePayload property\), 3308](#)
- [deploy_step\(\) \(in module ironic.drivers.base\), 3104](#)
- [deploy_template \(ironic.db.sqlalchemy.models.DeployTemplateStep attribute\), 1771](#)
- [deploy_template_id \(ironic.db.sqlalchemy.models.DeployTemplateStep attribute\), 1771](#)
- [DEPLOYDONE \(in module ironic.common.states\), 1321](#)
- [DEPLOYFAIL \(in module ironic.common.states\), 1321](#)

DEPLOYING (in module *ironic.common.states*), 1322

deploying_error_handler() (in module *ironic.conductor.utils*), 1580

DeployInterface (class in *ironic.drivers.base*), 2987

Deployment (class in *ironic.objects.deployment*), 3233

DeployTemplate (class in *ironic.db.sqlalchemy.models*), 1770

DeployTemplate (class in *ironic.objects.deploy_template*), 3217

DeployTemplateAlreadyExists, 1156

DeployTemplateCRUDNotification (class in *ironic.objects.deploy_template*), 3231

DeployTemplateCRUDPayload (class in *ironic.objects.deploy_template*), 3232

DeployTemplateDuplicateName, 1156

DeployTemplateNotFound, 1156

DeployTemplatesController (class in *ironic.api.controllers.v1.deploy_template*), 826

DeployTemplateStep (class in *ironic.db.sqlalchemy.models*), 1770

DEPLOYWAIT (in module *ironic.common.states*), 1322

description (*ironic.db.sqlalchemy.models.Chassis* attribute), 1768

description (*ironic.db.sqlalchemy.models.Node* attribute), 1773

description (*ironic.objects.chassis.Chassis* property), 3199

description (*ironic.objects.chassis.ChassisCRUDPayload* property), 3208

description (*ironic.objects.node.Node* property), 3264

description (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3291

description (*ironic.objects.node.NodeCRUDPayload* property), 3284

description (*ironic.objects.node.NodePayload* property), 3297

description (*ironic.objects.node.NodeSetPowerStatePayload* property), 3302

description (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3308

deserialize_context() (*ironic.common.rpc.RequestContextSerializer* method), 1309

deserialize_entity() (*ironic.common.rpc.RequestContextSerializer* method), 1310

destroy() (*ironic.objects.allocation.Allocation* method), 3155

destroy() (*ironic.objects.chassis.Chassis* method), 3199

destroy() (*ironic.objects.deploy_template.DeployTemplate* method), 3219

destroy() (*ironic.objects.deployment.Deployment* method), 3235

destroy() (*ironic.objects.node.Node* method), 3264

destroy() (*ironic.objects.port.Port* method), 3319

destroy() (*ironic.objects.portgroup.Portgroup* method), 3341

destroy() (*ironic.objects.trait.Trait* class method), 3359

destroy() (*ironic.objects.trait.TraitList* class method), 3364

destroy() (*ironic.objects.volume_connector.VolumeConnector* method), 3369

destroy() (*ironic.objects.volume_target.VolumeTarget* method), 3387

destroy_allocation() (*ironic.conductor.manager.ConductorManager* method), 1400

destroy_allocation() (*ironic.conductor.rpcapi.ConductorAPI* method), 1441

destroy_allocation() (*ironic.db.api.Connection* method), 1805

destroy_allocation() (*ironic.db.sqlalchemy.api.Connection* method), 1652

destroy_chassis() (*ironic.db.api.Connection* method), 1806

destroy_chassis() (*ironic.db.sqlalchemy.api.Connection* method), 1653

destroy_deploy_template() (*ironic.db.api.Connection* method), 1806

destroy_deploy_template() (*ironic.db.sqlalchemy.api.Connection* method), 1653

destroy_floppy_image_from_web_server() (in module *ironic.drivers.modules.ilo.common*), 2161

destroy_http_instance_images() (in mod-

- ule ironic.drivers.modules.deploy_utils*),
 2657
- destroy_images() (in module
ironic.drivers.modules.deploy_utils),
 2658
- destroy_node()
 (*ironic.conductor.manager.ConductorManager*
method), 1400
- destroy_node()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 1442
- destroy_node() (*ironic.db.api.Connection*
method), 1807
- destroy_node()
 (*ironic.db.sqlalchemy.api.Connection*
method), 1654
- destroy_port()
 (*ironic.conductor.manager.ConductorManager*
method), 1400
- destroy_port()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 1444
- destroy_port() (*ironic.db.api.Connection*
method), 1808
- destroy_port()
 (*ironic.db.sqlalchemy.api.Connection*
method), 1655
- destroy_portgroup()
 (*ironic.conductor.manager.ConductorManager*
method), 1400
- destroy_portgroup()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 1445
- destroy_portgroup()
 (*ironic.db.api.Connection method*),
 1808
- destroy_portgroup()
 (*ironic.db.sqlalchemy.api.Connection*
method), 1655
- destroy_volume_connector()
 (*ironic.conductor.manager.ConductorManager*
method), 1400
- destroy_volume_connector()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 1446
- destroy_volume_connector()
 (*ironic.db.api.Connection method*),
 1809
- destroy_volume_connector()
 (*ironic.db.sqlalchemy.api.Connection*
method), 1655
- destroy_volume_target()
 (*ironic.conductor.manager.ConductorManager*
method), 1400
- destroy_volume_target()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 1449
- destroy_volume_target()
 (*ironic.db.api.Connection method*),
 1810
- destroy_volume_target()
 (*ironic.db.sqlalchemy.api.Connection*
method), 1656
- detach_volumes() (in module
ironic.common.cinder), 1123
- detach_volumes()
 (*ironic.drivers.base.StorageInterface*
method), 3088
- detach_volumes()
 (*ironic.drivers.modules.fake.FakeStorage*
method), 2777
- detach_volumes()
 (*ironic.drivers.modules.storage.cinder.CinderStorage*
method), 2491
- detach_volumes()
 (*ironic.drivers.modules.storage.external.ExternalStorage*
method), 2497
- detach_volumes()
 (*ironic.drivers.modules.storage.noop.NoopStorage*
method), 2503
- detail() (*ironic.api.controllers.v1.chassis.ChassisController*
method), 813
- detail() (*ironic.api.controllers.v1.node.NodesController*
method), 878
- detail() (*ironic.api.controllers.v1.port.PortsController*
method), 905
- detail() (*ironic.api.controllers.v1.portgroup.PortgroupsController*
method), 916
- detect_vendor()
 (*ironic.drivers.base.ManagementInterface*
method), 3004
- detect_vendor()
 (*ironic.drivers.modules.ipmitool.IPMIManagement*
method), 2819
- detect_vendor()
 (*ironic.drivers.modules.redfish.management.RedfishManag*
method), 2409
- dhcp_options_for_instance() (in module
ironic.common.pxe_utils), 1278
- DHCPFactory (class in
ironic.common.dhcp_factory), 1132
- DHCPLoadError, 1156

- dict_valid() (in module *ironic.common.args*), 1096
- direct_deploy_should_convert_raw_image() (in module *ironic.drivers.modules.deploy_utils*), 2658
- DirectoryNotWritable, 1156
- disallowed_fields() (in module *ironic.api.controllers.v1.utils*), 983
- DISK (in module *ironic.common.boot_devices*), 1112
- DISK (in module *ironic.common.components*), 1129
- do_allocate() (in module *ironic.conductor.allocations*), 1375
- do_next_clean_step() (in module *ironic.conductor.cleaning*), 1384
- do_next_deploy_step() (in module *ironic.conductor.deployments*), 1389
- do_node_clean() (in module *ironic.conductor.cleaning*), 1385
- do_node_clean() (in module *ironic.conductor.manager.ConductorManager*), 1400
- do_node_clean() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1450
- do_node_clean_abort() (in module *ironic.conductor.cleaning*), 1387
- do_node_deploy() (in module *ironic.conductor.deployments*), 1390
- do_node_deploy() (in module *ironic.conductor.manager.ConductorManager*), 1401
- do_node_deploy() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1453
- do_node_rescue() (in module *ironic.conductor.manager.ConductorManager*), 1401
- do_node_rescue() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1456
- do_node_tear_down() (in module *ironic.conductor.manager.ConductorManager*), 1401
- do_node_tear_down() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1458
- do_node_unrescue() (in module *ironic.conductor.manager.ConductorManager*), 1401
- do_node_unrescue() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1460
- do_provisioning_action() (in module *ironic.conductor.manager.ConductorManager*), 1401
- do_provisioning_action() (in module *ironic.conductor.rpcapi.ConductorAPI*), 1462
- do_sync_power_state() (in module *ironic.conductor.manager*), 1408
- do_version_changes_for_db() (in module *ironic.objects.base.IronicObject*), 3172
- doc (in module *ironic.api.functions.FunctionDefinition*), 1066
- downgrade() (in module *ironic.db.sqlalchemy.migration*), 1762
- downgrade_lock() (in module *ironic.conductor.task_manager.TaskManager*), 1563
- download() (in module *ironic.common.glance_service.image_service.GlanceImageService*), 1086
- download() (in module *ironic.common.image_service.BaseImageService*), 1184
- download() (in module *ironic.common.image_service.FileImageService*), 1187
- download() (in module *ironic.common.image_service.HttpImageService*), 1191
- download_size() (in module *ironic.common.images*), 1212
- DracInspect (class in module *ironic.drivers.modules.drac.inspect*), 1975
- DracManagement (class in module *ironic.drivers.modules.drac.management*), 1985
- DracOperationError, 1157
- DracPower (class in module *ironic.drivers.modules.drac.power*), 2007
- DracRAID (class in module *ironic.drivers.modules.drac.raid*), 2016
- DracRedfishBIOS (class in module *ironic.drivers.modules.drac.bios*), 1943
- DracRedfishInspect (class in module *ironic.drivers.modules.drac.inspect*), 1976
- DracRedfishManagement (class in module *ironic.drivers.modules.drac.management*), 1985

1986
[attribute](#)), 1773
[DracRedfishPower](#) (class in [driver_internal_info](#) ([ironic.drivers.modules.drac.power](#)), ([ironic.objects.node.Node](#) property), 2008 3265)
[DracRedfishRAID](#) (class in [driver_internal_info](#) ([ironic.drivers.modules.drac.raid](#)), 2017 ([ironic.objects.node.NodeSetProvisionStatePayload](#) property), 3308)
[DracRedfishVendorPassthru](#) (class in [driver_passthru\(\)](#) (in module [ironic.drivers.modules.drac.vendor_passthru](#)), 2051 ([ironic.drivers.base](#)), 3109)
[DracRedfishVirtualMediaBoot](#) (class in [driver_validate\(\)](#) ([ironic.drivers.modules.drac.boot](#)), 1967 ([ironic.drivers.base.VendorInterface](#) method), 3092)
[DracVendorPassthru](#) (class in [driver_validate\(\)](#) ([ironic.drivers.modules.drac.vendor_passthru](#)), 2051 ([ironic.drivers.modules.noop.NoVendor](#) method), 2885)
[DracWSManBIOS](#) (class in [driver_vendor_passthru\(\)](#) ([ironic.drivers.modules.drac.bios](#)), 1944 ([ironic.conductor.manager.ConductorManager](#) method), 1401)
[DracWSManInspect](#) (class in [driver_vendor_passthru\(\)](#) ([ironic.drivers.modules.drac.inspect](#)), 1978 ([ironic.conductor.rpcapi.ConductorAPI](#) method), 1464)
[DracWSManManagement](#) (class in [DriverLoadError](#), 1157 ([ironic.drivers.modules.drac.management](#)), 1995)
[DracWSManPower](#) (class in [DriverNotFound](#), 1157 ([ironic.drivers.modules.drac.power](#)), 2009 [DriverNotFoundInEntrypoint](#), 1157 [DriverOperationError](#), 1157)
[DracWSManRAID](#) (class in [DriverPassthruController](#) (class in [ironic.drivers.modules.drac.raid](#)), 2022 ([ironic.api.controllers.v1.driver](#)), 833)
[DracWSManVendorPassthru](#) (class in [DriverRaidController](#) (class in [ironic.drivers.modules.drac.vendor_passthru](#)), 2053 ([ironic.api.controllers.v1.driver](#)), 835 [drivers](#) ([ironic.db.sqlalchemy.models.Conductor](#) attribute), 1768)
[driver](#) ([ironic.db.sqlalchemy.models.Node](#) attribute), 1773 [drivers](#) ([ironic.objects.conductor.Conductor](#) property), 3208)
[driver](#) ([ironic.objects.node.Node](#) property), 3265 [DriverPassthruController](#) (class in [ironic.objects.node.NodeCorrectedPowerStatePayload](#) property), 3291 ([ironic.api.controllers.v1.driver](#)), 837)
[driver](#) ([ironic.objects.node.NodeCRUDPayload](#) property), 3284 [dump_sdr\(\)](#) (in module [ironic.drivers.modules.ipmitool](#)), 2856
[driver](#) ([ironic.objects.node.NodePayload](#) property), 3297 [Duplicate](#), 1157
[driver](#) ([ironic.objects.node.NodeSetPowerStatePayload](#) property), 3302 [duplicate_steps\(\)](#) (in module [ironic.api.controllers.v1.deploy_template](#)), 831
[driver](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) property), 3308 [DuplicateName](#), 1157
E
[driver_info](#) ([ironic.db.sqlalchemy.models.Node](#) attribute), 1773 [eject_vmmedia\(\)](#) (in module [ironic.drivers.modules.redfish.boot](#)), 2402
[driver_info](#) ([ironic.objects.node.Node](#) property), 3265 [eject_vmmedia\(\)](#) ([ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru](#) method), 2487
[driver_info](#) ([ironic.objects.node.NodeCRUDPayload](#) property), 3284 [eject_vmmedia_devices\(\)](#) (in module [ironic.db.sqlalchemy.models.Node](#) attribute), 1773

- `ironic.drivers.modules.ilo.common`), 2162
- `emit()` (*ironic.objects.notification.NotificationBase* method), 3314
- `emit_console_notification()` (in module *ironic.conductor.notification_utils*), 1412
- `emit_end_notification()` (in module *ironic.api.controllers.v1.notification_utils*), 901
- `emit_power_set_notification()` (in module *ironic.conductor.notification_utils*), 1413
- `emit_power_state_corrected_notification()` (in module *ironic.conductor.notification_utils*), 1415
- `emit_provision_set_notification()` (in module *ironic.conductor.notification_utils*), 1416
- `emit_start_notification()` (in module *ironic.api.controllers.v1.notification_utils*), 902
- `enabled_supported_interfaces()` (in module *ironic.common.driver_factory*), 1146
- END (*ironic.objects.fields.NotificationStatus* attribute), 3247
- ENROLL (in module *ironic.common.states*), 1323
- `ensure_next_boot_device()` (in module *ironic.drivers.utils*), 3143
- `ensure_thread_contain_context()` (*ironic.common.context.RequestContext* method), 1130
- EnumField (class in *ironic.objects.fields*), 3242
- `erase_devices()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2204
- ERROR (in module *ironic.common.states*), 1324
- ERROR (*ironic.objects.fields.NotificationLevel* attribute), 3246
- ERROR (*ironic.objects.fields.NotificationStatus* attribute), 3247
- ESSENTIAL_PROPERTIES (*ironic.drivers.base.InspectInterface* attribute), 3000
- event (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3308
- event_type (*ironic.objects.allocation.AllocationCRUDNotification* property), 3166
- event_type (*ironic.objects.chassis.ChassisCRUDNotification* property), 3207
- event_type (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* property), 3231
- event_type (*ironic.objects.node.NodeConsoleNotification* property), 3288
- event_type (*ironic.objects.node.NodeCorrectedPowerStateNotification* property), 3289
- event_type (*ironic.objects.node.NodeCRUDNotification* property), 3281
- event_type (*ironic.objects.node.NodeMaintenanceNotification* property), 3295
- event_type (*ironic.objects.node.NodeSetPowerStateNotification* property), 3301
- event_type (*ironic.objects.node.NodeSetProvisionStateNotification* property), 3306
- event_type (*ironic.objects.notification.NotificationBase* property), 3315
- event_type (*ironic.objects.port.PortCRUDNotification* property), 3337
- event_type (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 3355
- event_type (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* property), 3384
- event_type (*ironic.objects.volume_target.VolumeTargetCRUDNotification* property), 3404
- events_valid() (in module *ironic.api.controllers.v1.event*), 843
- EventsController (class in *ironic.api.controllers.v1.event*), 842
- EventType (class in *ironic.objects.notification*), 3311
- ExclusiveLockRequired, 1157
- `execute()` (in module *ironic.common.utils*), 1344
- `execute_clean_step()` (in module *ironic.drivers.modules.agent_base*), 2580
- `execute_clean_step()` (*ironic.drivers.base.BaseInterface* method), 2965
- `execute_clean_step()` (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2564
- `execute_clean_step()` (*ironic.drivers.modules.agent_client.AgentClient* method), 2592
- `execute_clean_step()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeployMixin* method), 1935
- `execute_deploy_step()` (*ironic.drivers.base.BaseInterface* method), 2967
- `execute_deploy_step()` (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2564

- method), 2565
- execute_deploy_step() (ironic.drivers.modules.agent_client.AgentClient method), 2595
- execute_oem_manager_method() (in module ironic.drivers.modules.drac.utils), 2048
- execute_step() (in module ironic.drivers.modules.agent_base), 2580
- exists() (ironic.objects.trait.Trait class method), 3360
- export_configuration() (ironic.drivers.modules.drac.management.DracRedfishManagement method), 1987
- EXPORT_CONFIGURATION_ARGSINFO (ironic.drivers.modules.drac.management.DracRedfishManagement attribute), 1987
- expose() (in module ironic.api.method), 1080
- ExternalStorage (class in ironic.drivers.modules.storage.external), 2496
- extra (ironic.db.sqlalchemy.models.Allocation attribute), 1766
- extra (ironic.db.sqlalchemy.models.Chassis attribute), 1768
- extra (ironic.db.sqlalchemy.models.DeployTemplate attribute), 1770
- extra (ironic.db.sqlalchemy.models.Node attribute), 1773
- extra (ironic.db.sqlalchemy.models.Port attribute), 1777
- extra (ironic.db.sqlalchemy.models.Portgroup attribute), 1778
- extra (ironic.db.sqlalchemy.models.VolumeConnector attribute), 1779
- extra (ironic.db.sqlalchemy.models.VolumeTarget attribute), 1780
- extra (ironic.objects.allocation.Allocation property), 3156
- extra (ironic.objects.allocation.AllocationCRUDPayload property), 3167
- extra (ironic.objects.chassis.Chassis property), 3199
- extra (ironic.objects.chassis.ChassisCRUDPayload property), 3208
- extra (ironic.objects.deploy_template.DeployTemplate property), 3220
- extra (ironic.objects.deploy_template.DeployTemplateCRUDPayload property), 3232
- extra (ironic.objects.node.Node property), 3265
- extra (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3291
- extra (ironic.objects.node.NodeCRUDPayload property), 3284
- extra (ironic.objects.node.NodePayload property), 3297
- extra (ironic.objects.node.NodeSetPowerStatePayload property), 3302
- extra (ironic.objects.node.NodeSetProvisionStatePayload property), 3308
- extra (ironic.objects.port.Port property), 3320
- extra (ironic.objects.port.PortCRUDPayload property), 3338
- extra (ironic.objects.portgroup.Portgroup property), 3342
- extra (ironic.objects.portgroup.PortgroupCRUDPayload property), 3356
- extra (ironic.objects.volume_connector.VolumeConnector property), 3370
- extra (ironic.objects.volume_connector.VolumeConnectorCRUDPayload property), 3385
- extra (ironic.objects.volume_target.VolumeTarget property), 3388
- extra (ironic.objects.volume_target.VolumeTargetCRUDPayload property), 3405
- extra_options (ironic.api.functions.FunctionDefinition attribute), 1066
- ## F
- factory_reset() (ironic.drivers.base.BIOSInterface method), 2955
- factory_reset() (ironic.drivers.modules.drac.bios.DracWSManBIOS method), 1947
- factory_reset() (ironic.drivers.modules.fake.FakeBIOS method), 2700
- factory_reset() (ironic.drivers.modules.ilo.bios.IloBIOS method), 2108
- factory_reset() (ironic.drivers.modules.irmc.bios.IRMCBIOS method), 2263
- factory_reset() (ironic.drivers.modules.noop.NoBIOS method), 2868
- factory_reset() (ironic.drivers.modules.redfish.bios.RedfishBIOS method), 2382
- fail_on_error() (in module ironic.conductor.utils), 1581

[FailedToCleanDHCPOpts](#), 1158
[FailedToGetIPAddressOnPort](#), 1158
[FailedToGetSensorData](#), 1158
[FailedToParseSensorData](#), 1158
[FailedToUpdateDHCPoptOnPort](#), 1158
[FailedToUpdateMacOnPort](#), 1158
[FailMixin](#) (class in *ironic.drivers.modules.noop*), 2863
[FakeBIOS](#) (class in *ironic.drivers.modules.fake*), 2696
[FakeBoot](#) (class in *ironic.drivers.modules.fake*), 2705
[FakeConsole](#) (class in *ironic.drivers.modules.fake*), 2712
[FakeDeploy](#) (class in *ironic.drivers.modules.fake*), 2718
[FakeHardware](#) (class in *ironic.drivers.fake_hardware*), 3112
[FakeInspect](#) (class in *ironic.drivers.modules.fake*), 2728
[FakeManagement](#) (class in *ironic.drivers.modules.fake*), 2733
[FakePower](#) (class in *ironic.drivers.modules.fake*), 2757
[FakeRAID](#) (class in *ironic.drivers.modules.fake*), 2765
[FakeRescue](#) (class in *ironic.drivers.modules.fake*), 2770
[FakeStorage](#) (class in *ironic.drivers.modules.fake*), 2776
[FakeVendorA](#) (class in *ironic.drivers.modules.fake*), 2781
[FakeVendorB](#) (class in *ironic.drivers.modules.fake*), 2784
[fast_track_able\(\)](#) (in module *ironic.conductor.utils*), 1581
[FASTTRACK_LOOKUP_ALLOWED_STATES](#) (in module *ironic.common.states*), 1324
[fault](#) (*ironic.db.sqlalchemy.models.Node* attribute), 1773
[fault](#) (*ironic.objects.node.Node* property), 3265
[fault](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 3291
[fault](#) (*ironic.objects.node.NodeCRUDPayload* property), 3284
[fault](#) (*ironic.objects.node.NodePayload* property), 3297
[fault](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 3302
[fault](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3308
[faultstring](#) (*ironic.common.exception.ClientSideError* property), 1154
[faultstring](#) (*ironic.common.exception.InvalidInput* property), 1164
[faultstring](#) (*ironic.common.exception.UnknownArgument* property), 1173
[faultstring](#) (*ironic.common.exception.UnknownAttribute* property), 1174
[fetch\(\)](#) (in module *ironic.common.images*), 1212
[fetch_image\(\)](#) (*ironic.drivers.modules.image_cache.ImageCache* method), 2789
[fetch_images\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 2659
[fetch_into\(\)](#) (in module *ironic.common.images*), 1212
[fields](#) (*ironic.objects.allocation.Allocation* attribute), 3156
[fields](#) (*ironic.objects.allocation.AllocationCRUDNotification* attribute), 3166
[fields](#) (*ironic.objects.allocation.AllocationCRUDPayload* attribute), 3167
[fields](#) (*ironic.objects.base.IronicObject* attribute), 3175
[fields](#) (*ironic.objects.bios.BIOSSetting* attribute), 3185
[fields](#) (*ironic.objects.bios.BIOSSettingList* attribute), 3193
[fields](#) (*ironic.objects.chassis.Chassis* attribute), 3199
[fields](#) (*ironic.objects.chassis.ChassisCRUDNotification* attribute), 3207
[fields](#) (*ironic.objects.chassis.ChassisCRUDPayload* attribute), 3208
[fields](#) (*ironic.objects.conductor.Conductor* attribute), 3208
[fields](#) (*ironic.objects.deploy_template.DeployTemplate* attribute), 3220
[fields](#) (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* attribute), 3231
[fields](#) (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 3232
[fields](#) (*ironic.objects.deployment.Deployment* attribute), 3236
[fields](#) (*ironic.objects.node.Node* attribute), 3265
[fields](#) (*ironic.objects.node.NodeConsoleNotification* attribute), 3288
[fields](#) (*ironic.objects.node.NodeCorrectedPowerStateNotification* attribute), 3289
[fields](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 3291

fields (*ironic.objects.node.NodeCRUDNotification* attribute), 3281 *ironic.common.utils*), 1346

fields (*ironic.objects.node.NodeCRUDPayload* attribute), 3284 *file_mime_type()* (in module *ironic.common.utils*), 1347

fields (*ironic.objects.node.NodeMaintenanceNotification* attribute), 3295 *FileImageService* (class in *ironic.common.image_service*), 1186

fields (*ironic.objects.node.NodePayload* attribute), 3297 *FileSystemNotSupported*, 1158

fields (*ironic.objects.node.NodeSetPowerStateNotification* attribute), 3301 *filter_target_raid_config()* (in module *ironic.common.raid*), 1300

fields (*ironic.objects.node.NodeSetPowerStatePayload* attribute), 3302 *finalize_rescue()* (in module *ironic.drivers.modules.agent_client.AgentClient*), 2599

fields (*ironic.objects.node.NodeSetProvisionStateNotification* attribute), 3306 *find_step()* (in module *ironic.conductor.steps*), 548

fields (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 3308 *find_step()* (in module *ironic.drivers.modules.agent_base*), 2582

fields (*ironic.objects.notification.EventType* attribute), 3312 *FirmwareImageLocation* (class in *ironic.drivers.modules.ilo.firmware_processor*), 2194

fields (*ironic.objects.notification.NotificationBase* attribute), 3315 *FirmwareProcessor* (class in *ironic.drivers.modules.ilo.firmware_processor*), 2195

fields (*ironic.objects.notification.NotificationPayloadBase* attribute), 3316 *first_method()* (in module *ironic.drivers.modules.fake.FakeVendorA*), 2782

fields (*ironic.objects.notification.NotificationPublisher* attribute), 3317 *flash_firmware_sum()* (in module *ironic.drivers.modules.ilo.management.IloManagement*), 2210

fields (*ironic.objects.port.Port* attribute), 3320 *FlatNetwork* (class in *ironic.drivers.modules.network.flat*), 2348

fields (*ironic.objects.port.PortCRUDNotification* attribute), 3337 *FlexibleDict* (class in *ironic.objects.fields*), 3242

fields (*ironic.objects.port.PortCRUDPayload* attribute), 3338 *FlexibleDictField* (class in *ironic.objects.fields*), 3244

fields (*ironic.objects.portgroup.Portgroup* attribute), 3342 *FLOPPY* (in module *ironic.common.boot_devices*), 1112

fields (*ironic.objects.portgroup.PortgroupCRUDNotification* attribute), 3355 *Forbidden*, 1159

fields (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 3356 *force_persistent_boot()* (in module *ironic.drivers.utils*), 3144

fields (*ironic.objects.trait.Trait* attribute), 3361 *force_CRUD_notification_convert()* (in module *ironic.common.images*), 1212

fields (*ironic.objects.trait.TraitList* attribute), 3365 *format_CRUD_payload()* (in module *ironic.api*), 1080

fields (*ironic.objects.volume_connector.VolumeConnector* attribute), 3370 *fourth_method_shared_lock()* (in module *ironic.drivers.modules.fake.FakeVendorB*), 2785

fields (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* attribute), 3384 *from_chassis* (*ironic.api.controllers.v1.node.NodesController*), 884

fields (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 3385 *from_power* (*ironic.objects.node.NodeCorrectedPowerStatePayload*), 884

fields (*ironic.objects.volume_target.VolumeTarget* attribute), 3388 *file_has_content()* (in module *ironic.common.utils*), 1346

fields (*ironic.objects.volume_target.VolumeTargetCRUDNotification* attribute), 3405 *file_has_content()* (in module *ironic.common.utils*), 1346

fields (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 3405 *file_has_content()* (in module *ironic.common.utils*), 1346

file_has_content() (in module *ironic.common.utils*), 1346

FSM (class in *ironic.common.fsm*), 1177

FunctionArgument (class in *ironic.api.functions*), 1064

FunctionDefinition (class in *ironic.api.functions*), 1065

G

GenericHardware (class in *ironic.drivers.generic*), 3116

get() (*ironic.api.controllers.v1.node.BootDeviceController* method), 822

get() (*ironic.api.controllers.v1.node.NodeConsoleController* method), 853

get() (*ironic.api.controllers.v1.node.NodeStatesController* method), 857

get() (*ironic.api.controllers.v1.volume.VolumeController* method), 1023

get() (*ironic.api.functions.FunctionDefinition* static method), 1066

get() (*ironic.drivers.modules.snmp.SNMPClient* method), 2925

get() (*ironic.objects.allocation.Allocation* class method), 3156

get() (*ironic.objects.bios.BIOSSetting* class method), 3186

get() (*ironic.objects.chassis.Chassis* class method), 3199

get() (*ironic.objects.node.Node* class method), 3267

get() (*ironic.objects.port.Port* class method), 3320

get() (*ironic.objects.portgroup.Portgroup* class method), 3342

get() (*ironic.objects.volume_connector.VolumeConnector* class method), 3370

get() (*ironic.objects.volume_target.VolumeTarget* class method), 3389

get_active_hardware_type_dict() (*ironic.db.api.Connection* method), 1811

get_active_hardware_type_dict() (*ironic.db.sqlalchemy.api.Connection* method), 1657

get_adapter() (in *ironic.common.keystone*), 1217

get_admin_context() (in *ironic.common.context*), 1132

get_agent_iso() (in *ironic.drivers.utils*), 3145

get_agent_kernel_ramdisk() (in *ironic.drivers.utils*), 3145

get_all() (*ironic.api.controllers.v1.allocation.AllocationsController* method), 805

get_all() (*ironic.api.controllers.v1.allocation.NodeAllocationController* method), 811

get_all() (*ironic.api.controllers.v1.bios.NodeBiosController* method), 811

get_all() (*ironic.api.controllers.v1.chassis.ChassisController* method), 815

get_all() (*ironic.api.controllers.v1.conductor.CollectorsController* method), 822

get_all() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 826

get_all() (*ironic.api.controllers.v1.driver.DriversController* method), 837

get_all() (*ironic.api.controllers.v1.node.IndicatorController* method), 847

get_all() (*ironic.api.controllers.v1.node.NodesController* method), 885

get_all() (*ironic.api.controllers.v1.node.NodeTraitsController* method), 873

get_all() (*ironic.api.controllers.v1.node.NodeVIFController* method), 875

get_all() (*ironic.api.controllers.v1.port.PortsController* method), 909

get_all() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 919

get_all() (*ironic.api.controllers.v1.ramdisk.LookupController* method), 928

get_all() (*ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 1026

get_all() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 1041

get_allocation_by_id() (*ironic.db.api.Connection* method), 1812

get_allocation_by_id() (*ironic.db.sqlalchemy.api.Connection* method), 1659

get_allocation_by_name() (*ironic.db.api.Connection* method), 1813

get_allocation_by_name() (*ironic.db.sqlalchemy.api.Connection* method), 1659

get_allocation_by_uuid() (*ironic.db.api.Connection* method), 1814

get_allocation_by_uuid() (*ironic.db.sqlalchemy.api.Connection* method), 1660

get_allocation_list()

- (*ironic.db.api.Connection* method), 1814
- `get_allocation_list()` (*ironic.db.sqlalchemy.api.Connection* method), 1661
- `get_allowed_exmods()` (in module *ironic.common.rpc*), 1313
- `get_and_validate_firmware_image_info()` (in module *ironic.drivers.modules.ilo.firmware_processor*), 2197
- `get_arg()` (*ironic.api.functions.FunctionDefinition* method), 1067
- `get_async_step_return_state()` (in module *ironic.drivers.modules.deploy_utils*), 2660
- `get_attached_vif()` (in module *ironic.conductor.utils*), 1583
- `get_auth()` (in module *ironic.common.keystone*), 1218
- `get_backend()` (in module *ironic.db.migration*), 1911
- `get_backend()` (in module *ironic.db.sqlalchemy.api*), 1761
- `get_bios_config()` (*ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthru* method), 2056
- `get_bios_setting()` (*ironic.db.api.Connection* method), 1817
- `get_bios_setting()` (*ironic.db.sqlalchemy.api.Connection* method), 1663
- `get_bios_setting_list()` (*ironic.db.api.Connection* method), 1818
- `get_bios_setting_list()` (*ironic.db.sqlalchemy.api.Connection* method), 1664
- `get_boot_device()` (*ironic.conductor.manager.ConductorManager* method), 1401
- `get_boot_device()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1470
- `get_boot_device()` (*ironic.drivers.base.ManagementInterface* method), 3006
- `get_boot_device()` (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 1996
- `get_boot_device()` (*ironic.drivers.modules.fake.FakeManagement* method), 2733
- `get_boot_device()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 2063
- `get_boot_device()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2211
- `get_boot_device()` (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 2821
- `get_boot_device()` (*ironic.drivers.modules.noop_mgmt.NoopManagement* method), 2888
- `get_boot_device()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 2411
- `get_boot_device()` (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 2512
- `get_boot_mode()` (in module *ironic.drivers.modules.boot_mode_utils*), 2634
- `get_boot_mode()` (*ironic.drivers.base.ManagementInterface* method), 2068
- `get_boot_mode()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 2066
- `get_boot_mode()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2214
- `get_boot_mode()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 2413
- `get_boot_mode_for_deploy()` (in module *ironic.drivers.modules.boot_mode_utils*), 2635
- `get_boot_option()` (in module *ironic.drivers.modules.deploy_utils*), 2661
- `get_by_address()` (*ironic.objects.port.Port* class method), 3322
- `get_by_address()` (*ironic.objects.portgroup.Portgroup* class method), 3343
- `get_by_hostname()` (*ironic.objects.conductor.Conductor* class method), 3209
- `get_by_id()` (*ironic.objects.allocation.Allocation* class method), 3157

[get_by_id\(\) \(ironic.objects.chassis.Chassis class method\), 3200](#)
[get_by_id\(\) \(ironic.objects.deploy_template.DeployTemplate class method\), 3220](#)
[get_by_id\(\) \(ironic.objects.node.Node class method\), 3267](#)
[get_by_id\(\) \(ironic.objects.port.Port class method\), 3323](#)
[get_by_id\(\) \(ironic.objects.portgroup.Portgroup class method\), 3344](#)
[get_by_id\(\) \(ironic.objects.volume_connector.VolumeConnector class method\), 3372](#)
[get_by_id\(\) \(ironic.objects.volume_target.VolumeTarget class method\), 3391](#)
[get_by_instance_uuid\(\) \(ironic.objects.node.Node class method\), 3268](#)
[get_by_name\(\) \(ironic.objects.allocation.Allocation class method\), 3158](#)
[get_by_name\(\) \(ironic.objects.deploy_template.DeployTemplate class method\), 3222](#)
[get_by_name\(\) \(ironic.objects.node.Node class method\), 3269](#)
[get_by_name\(\) \(ironic.objects.port.Port class method\), 3324](#)
[get_by_name\(\) \(ironic.objects.portgroup.Portgroup class method\), 3345](#)
[get_by_node_id\(\) \(ironic.objects.bios.BIOSSettingList class method\), 3193](#)
[get_by_node_id\(\) \(ironic.objects.trait.TraitList class method\), 3365](#)
[get_by_node_uuid\(\) \(ironic.objects.deployment.Deployment class method\), 3236](#)
[get_by_port_addresses\(\) \(ironic.objects.node.Node class method\), 3270](#)
[get_by_uuid\(\) \(ironic.objects.allocation.Allocation class method\), 3159](#)
[get_by_uuid\(\) \(ironic.objects.chassis.Chassis class method\), 3201](#)
[get_by_uuid\(\) \(ironic.objects.deploy_template.DeployTemplate class method\), 3223](#)
[get_by_uuid\(\) \(ironic.objects.deployment.Deployment class method\), 3237](#)
[get_by_uuid\(\) \(ironic.objects.node.Node class method\), 3271](#)
[get_by_uuid\(\) \(ironic.objects.port.Port class method\), 3325](#)
[get_by_uuid\(\) \(ironic.objects.portgroup.Portgroup class method\), 3346](#)
[get_by_uuid\(\) \(ironic.objects.volume_connector.VolumeConnector class method\), 3374](#)
[get_by_uuid\(\) \(ironic.objects.volume_target.VolumeTarget class method\), 3392](#)
[get_chassis_by_id\(\) \(ironic.db.api.Connection method\), 1818](#)
[get_chassis_by_id\(\) \(ironic.db.sqlalchemy.api.Connection method\), 1665](#)
[get_chassis_by_uuid\(\) \(ironic.db.api.Connection method\), 1819](#)
[get_chassis_by_uuid\(\) \(ironic.db.sqlalchemy.api.Connection method\), 1666](#)
[get_chassis_list\(\) \(ironic.db.api.Connection method\), 1819](#)
[get_chassis_list\(\) \(ironic.db.sqlalchemy.api.Connection method\), 1666](#)
[get_class\(\) \(in module ironic.db.sqlalchemy.models\), 1780](#)
[get_clean_steps\(\) \(ironic.drivers.base.BaseInterface method\), 2969](#)
[get_clean_steps\(\) \(ironic.drivers.modules.agent.AgentRAID method\), 2531](#)
[get_clean_steps\(\) \(ironic.drivers.modules.agent_base.AgentDeployMixin method\), 2567](#)
[get_clean_steps\(\) \(ironic.drivers.modules.agent_client.AgentClient method\), 2601](#)
[get_clean_steps\(\) \(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method\), 1935](#)
[get_cleaning_network_uuid\(\) \(ironic.common.neutron.NeutronNetworkInterfaceMixin method\), 1233](#)
[get_deployments\(\) \(in module ironic.common.cinder\), 1126](#)
[get_client\(\) \(in module ironic.common.neutron\), 1237](#)
[get_client\(\) \(in module ironic.common.rpc\), 1313](#)
[get_command_error\(\) \(in module ironic.drivers.modules.agent_client\), 2625](#)

`get_commands_status()` (*ironic.drivers.modules.agent_client.AgentClient* method), 2340
method), 2604
`get_conductor()` (*ironic.db.api.Connection* method), 1821
`get_conductor()` (*ironic.db.sqlalchemy.api.Connection* method), 1667
`get_conductor_for()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1473
`get_conductor_list()` (*ironic.db.api.Connection* method), 1822
`get_conductor_list()` (*ironic.db.sqlalchemy.api.Connection* method), 1669
`get_config()` (in module *ironic.drivers.modules.drac.bios*), 1953
`get_configdrive_image()` (in module *ironic.conductor.utils*), 1584
`get_configuration()` (in module *ironic.common.molds*), 1223
`get_console()` (*ironic.drivers.base.ConsoleInterface* method), 2984
`get_console()` (*ironic.drivers.modules.fake.FakeConsole* method), 2713
`get_console()` (*ironic.drivers.modules.ipmitool.IPMIShellin* method), 2841
`get_console()` (*ironic.drivers.modules.ipmitool.IPMIShellin* method), 2845
`get_console()` (*ironic.drivers.modules.noop.NoConsole* method), 2871
`get_console_information()` (*ironic.conductor.manager.ConductorManager* method), 1401
`get_console_information()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1474
`get_controller_reserved_names()` (in module *ironic.api.controllers.v1.utils*), 983
`get_current_boot_mode()` (in module *ironic.drivers.modules.ilo.common*), 2163
`get_current_topic()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1475
`get_current_vif()` (*ironic.drivers.base.NetworkInterface* method), 3047
`get_current_vif()` (*ironic.drivers.modules.network.common.VIFPortIDMixin* method), 2340
`get_current_vif()` (*ironic.drivers.modules.network.noop.NoopNetwork* method), 2371
`get_default_boot_option()` (in module *ironic.drivers.modules.deploy_utils*), 2662
`get_deploy_steps()` (*ironic.drivers.base.BaseInterface* method), 2971
`get_deploy_steps()` (*ironic.drivers.modules.agent.AgentRAID* method), 2532
`get_deploy_steps()` (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2568
`get_deploy_steps()` (*ironic.drivers.modules.agent_client.AgentClient* method), 2611
`get_deploy_template_by_id()` (*ironic.db.api.Connection* method), 1824
`get_deploy_template_by_id()` (*ironic.db.sqlalchemy.api.Connection* method), 1670
`get_deploy_template_by_name()` (*ironic.db.api.Connection* method), 1825
`get_deploy_template_by_name()` (*ironic.db.sqlalchemy.api.Connection* method), 1671
`get_deploy_template_by_uuid()` (*ironic.db.api.Connection* method), 1826
`get_deploy_template_by_uuid()` (*ironic.db.sqlalchemy.api.Connection* method), 1672
`get_deploy_template_list()` (*ironic.db.api.Connection* method), 1827
`get_deploy_template_list()` (*ironic.db.sqlalchemy.api.Connection* method), 1673
`get_deploy_template_list_by_names()` (*ironic.db.api.Connection* method), 1828
`get_deploy_template_list_by_names()` (*ironic.db.sqlalchemy.api.Connection* method), 1675
`get_disk_label()` (in module *ironic.drivers.modules.deploy_utils*),

- 2663
- `get_drac_client()` (in module `ironic.drivers.modules.drac.common`), 1972
- `get_driver()` (`ironic.common.driver_factory.BaseDriverFactory` method), 1136
- `get_driver_properties()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_driver_properties()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1476
- `get_driver_vendor_passthru_methods()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_driver_vendor_passthru_methods()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1477
- `get_enabled_macs()` (in module `ironic.drivers.modules.redfish.utils`), 2479
- `get_endpoint()` (in module `ironic.common.keystone`), 1219
- `get_enforcer()` (in module `ironic.common.policy`), 1264
- `get_file_path_from_label()` (in module `ironic.common.pxe_utils`), 1280
- `get_free_port_like_object()` (in module `ironic.drivers.modules.network.common`), 2342
- `get_hardware_type()` (in module `ironic.common.driver_factory`), 1147
- `get_http_url_path_from_label()` (in module `ironic.common.pxe_utils`), 1281
- `get_ilo_object()` (in module `ironic.drivers.modules.ilo.common`), 2164
- `get_image_download_source()` (in module `ironic.drivers.modules.deploy_utils`), 2664
- `get_image_info()` (in module `ironic.common.pxe_utils`), 1283
- `get_image_instance_info()` (in module `ironic.drivers.modules.deploy_utils`), 2664
- `get_image_properties()` (in module `ironic.common.images`), 1212
- `get_image_service()` (in module `ironic.common.image_service`), 1195
- `get_indicator_state()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_indicator_state()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1479
- `get_indicator_state()` (`ironic.drivers.base.ManagementInterface` method), 3010
- `get_indicator_state()` (`ironic.drivers.modules.fake.FakeManagement` method), 2735
- `get_indicator_state()` (`ironic.drivers.modules.redfish.management.RedfishManagement` method), 2415
- `get_inspection_network_uuid()` (`ironic.common.neutron.NeutronNetworkInterfaceMixin` method), 1233
- `get_instance()` (in module `ironic.db.api`), 1910
- `get_instance_image_info()` (in module `ironic.common.pxe_utils`), 1285
- `get_interface()` (in module `ironic.common.driver_factory`), 1148
- `get_ip_addresses()` (`ironic.dhcp.base.BaseDHCP` method), 1912
- `get_ip_addresses()` (`ironic.dhcp.neutron.NeutronDHCPApi` method), 1919
- `get_ip_addresses()` (`ironic.dhcp.none.NoneDHCPApi` method), 1926
- `get_ipxe_boot_file()` (in module `ironic.drivers.modules.deploy_utils`), 2666
- `get_ipxe_config_template()` (in module `ironic.drivers.modules.deploy_utils`), 2667
- `get_ipxe_root_dir()` (in module `ironic.common.pxe_utils`), 1287
- `get_irmc_client()` (in module `ironic.drivers.modules.irmc.common`), 2284
- `get_irmc_report()` (in module `ironic.drivers.modules.irmc.common`), 2285
- `get_ironic_api_url()` (in module `ironic.drivers.modules.deploy_utils`), 2668
- `get_job()` (in module `ironic.drivers.modules.drac.job`), 1981
- `get_kernel_append_params()` (in module `ironic.drivers.utils`), 3146

- `get_kernel_ramdisk_info()` (in module `ironic.common.pxe_utils`), 1287
- `get_last_command_status()` (`ironic.drivers.modules.agent_client.AgentClient` method), 2613
- `get_local_group_information()` (in module `ironic.common.neutron`), 1237
- `get_logical_disk_properties()` (in module `ironic.common.raid`), 1302
- `get_logical_disk_properties()` (`ironic.drivers.base.RAIDInterface` method), 3078
- `get_logical_disks()` (`ironic.drivers.modules.drac.raid.DracWSMgmtRAID` method), 2029
- `get_mac_addresses()` (`ironic.drivers.base.ManagementInterface` method), 3012
- `get_mac_addresses()` (`ironic.drivers.modules.redfish.management.RedfishManagementClient` method), 2417
- `get_neutron_port_data()` (in module `ironic.common.neutron`), 1240
- `get_next()` (in module `ironic.api.controllers.v1.collection`), 819
- `get_next()` (`ironic.drivers.modules.snmp.SNMPClient` method), 2926
- `get_node_by_id()` (`ironic.db.api.Connection` method), 1829
- `get_node_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 1676
- `get_node_by_instance()` (`ironic.db.api.Connection` method), 1829
- `get_node_by_instance()` (`ironic.db.sqlalchemy.api.Connection` method), 1676
- `get_node_by_name()` (`ironic.db.api.Connection` method), 1830
- `get_node_by_name()` (`ironic.db.sqlalchemy.api.Connection` method), 1677
- `get_node_by_port_addresses()` (`ironic.db.api.Connection` method), 1831
- `get_node_by_port_addresses()` (`ironic.db.sqlalchemy.api.Connection` method), 1677
- `get_node_by_uuid()` (`ironic.db.api.Connection` method), 1831
- `get_node_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 1678
- `get_node_capability()` (in module `ironic.drivers.utils`), 3147
- `get_node_list()` (`ironic.db.api.Connection` method), 1832
- `get_node_list()` (`ironic.db.sqlalchemy.api.Connection` method), 1678
- `get_node_mac_addresses()` (in module `ironic.drivers.utils`), 3148
- `get_node_network_data()` (`ironic.drivers.base.NetworkInterface` method), 3048
- `get_node_network_data()` (`ironic.drivers.modules.network.common.NeutronVIFPort` method), 2331
- `get_node_network_clean_steps()` (in module `ironic.conductor.utils`), 1585
- `get_node_next_deploy_steps()` (in module `ironic.conductor.utils`), 1585
- `get_node_portmap()` (in module `ironic.common.neutron`), 1242
- `get_node_tags_by_node_id()` (`ironic.db.api.Connection` method), 1834
- `get_node_tags_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 1681
- `get_node_traits_by_node_id()` (`ironic.db.api.Connection` method), 1835
- `get_node_traits_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 1681
- `get_node_vendor_passthru_methods()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_node_vendor_passthru_methods()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1482
- `get_node_vif_ids()` (in module `ironic.common.network`), 1226
- `get_node_with_token()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_node_with_token()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1483

[get_nodeinfo_list\(\)](#) (*ironic.db.api.Connection* method), 1836
[get_nodeinfo_list\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1682
[get_nodes_controller_reserved_names\(\)](#) (in module *ironic.api.controllers.v1.node*), 897
[get_not_versions\(\)](#) (*ironic.db.api.Connection* method), 1839
[get_not_versions\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1686
[get_object_versions\(\)](#) (in module *ironic.common.release_mappings*), 1307
[get_offline_conductors\(\)](#) (*ironic.db.api.Connection* method), 1841
[get_offline_conductors\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1688
[get_one\(\)](#) (*ironic.api.controllers.v1.allocation.AllocationsController* method), 808
[get_one\(\)](#) (*ironic.api.controllers.v1.bios.NodeBiosController* method), 812
[get_one\(\)](#) (*ironic.api.controllers.v1.chassis.ChassisController* method), 816
[get_one\(\)](#) (*ironic.api.controllers.v1.conductor.ConductorsController* method), 824
[get_one\(\)](#) (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 829
[get_one\(\)](#) (*ironic.api.controllers.v1.driver.DriversController* method), 837
[get_one\(\)](#) (*ironic.api.controllers.v1.node.IndicatorController* method), 848
[get_one\(\)](#) (*ironic.api.controllers.v1.node.NodesController* method), 892
[get_one\(\)](#) (*ironic.api.controllers.v1.port.PortsController* method), 912
[get_one\(\)](#) (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 921
[get_one\(\)](#) (*ironic.api.controllers.v1.volume_connections.ConnectionsController* method), 1030
[get_one\(\)](#) (*ironic.api.controllers.v1.volume_target.TargetController* method), 1045
[get_online_conductors\(\)](#) (*ironic.db.api.Connection* method), 1842
[get_online_conductors\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1689
[get_oslo_policy_enforcer\(\)](#) (in module *ironic.common.policy*), 1264
[get_partition_uuids\(\)](#) (*ironic.drivers.modules.agent_client.AgentClient* method), 2614
[get_patch_values\(\)](#) (in module *ironic.api.controllers.v1.utils*), 984
[get_path_relative_to_tftp_root\(\)](#) (in module *ironic.common.pxe_utils*), 1289
[get_pecan_config\(\)](#) (in module *ironic.api.app*), 1064
[get_physical_disks\(\)](#) (in module *ironic.drivers.modules.redfish.raid*), 2478
[get_physnets_by_port_uuid\(\)](#) (in module *ironic.common.neutron*), 1244
[get_physnets_by_portgroup_id\(\)](#) (in module *ironic.common.network*), 1228
[get_physnets_for_node\(\)](#) (in module *ironic.common.network*), 1230
[get_port_by_address\(\)](#) (*ironic.db.api.Connection* method), 1842
[get_port_by_address\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1689
[get_port_by_id\(\)](#) (*ironic.db.api.Connection* method), 1843
[get_port_by_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1690
[get_port_by_name\(\)](#) (*ironic.db.api.Connection* method), 1843
[get_port_by_name\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1690
[get_port_by_uuid\(\)](#) (*ironic.db.api.Connection* method), 1844
[get_port_by_uuid\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1690
[get_volume_targets\(\)](#) (*ironic.db.api.Connection* method), 1844
[get_volume_targets\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1691
[get_portgroup_by_address\(\)](#) (*ironic.db.api.Connection* method), 1845

- `get_portgroup_by_address()`
(*ironic.db.sqlalchemy.api.Connection* method), 1692
- `get_portgroup_by_id()` (in module *ironic.common.network*), 1231
- `get_portgroup_by_id()`
(*ironic.db.api.Connection* method), 1846
- `get_portgroup_by_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 1693
- `get_portgroup_by_name()`
(*ironic.db.api.Connection* method), 1847
- `get_portgroup_by_name()`
(*ironic.db.sqlalchemy.api.Connection* method), 1694
- `get_portgroup_by_uuid()`
(*ironic.db.api.Connection* method), 1848
- `get_portgroup_by_uuid()`
(*ironic.db.sqlalchemy.api.Connection* method), 1695
- `get_portgroup_list()`
(*ironic.db.api.Connection* method), 1848
- `get_portgroup_list()`
(*ironic.db.sqlalchemy.api.Connection* method), 1695
- `get_portgroups_by_node_id()`
(*ironic.db.api.Connection* method), 1850
- `get_portgroups_by_node_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 1697
- `get_ports_by_node_id()`
(*ironic.db.api.Connection* method), 1852
- `get_ports_by_node_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 1699
- `get_ports_by_portgroup_id()` (in module *ironic.common.network*), 1231
- `get_ports_by_portgroup_id()`
(*ironic.db.api.Connection* method), 1854
- `get_ports_by_portgroup_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 1701
- `get_power_state()`
(*ironic.drivers.base.PowerInterface* method), 3064
- `get_power_state()`
(*ironic.drivers.modules.agent_power.AgentPower* method), 2626
- `get_power_state()`
(*ironic.drivers.modules.drac.power.DracWSManPower* method), 2010
- `get_power_state()`
(*ironic.drivers.modules.fake.FakePower* method), 2758
- `get_power_state()`
(*ironic.drivers.modules.ibmcpower.IBMCPower* method), 2080
- `get_power_state()`
(*ironic.drivers.modules.ilo.power.IloPower* method), 2240
- `get_power_state()`
(*ironic.drivers.modules.ipmitool.IPMIPower* method), 2832
- `get_power_state()`
(*ironic.drivers.modules.irmc.power.IRMCPower* method), 2316
- `get_power_state()`
(*ironic.drivers.modules.redfish.power.RedfishPower* method), 2448
- `get_power_state()`
(*ironic.drivers.modules.snmp.SNMPPower* method), 2942
- `get_power_state()`
(*ironic.drivers.modules.xclarity.power.XClarityPower* method), 2518
- `get_properties()` (in module *ironic.drivers.modules.xclarity.common*), 2507
- `get_properties()`
(*ironic.drivers.base.BareDriver* method), 2960
- `get_properties()`
(*ironic.drivers.base.BaseInterface* method), 2972
- `get_properties()`
(*ironic.drivers.base.NetworkInterface* method), 3050
- `get_properties()`
(*ironic.drivers.base.RAIDInterface* method), 3079
- `get_properties()`
(*ironic.drivers.hardware_type.AbstractHardwareType* method), 3121
- `get_properties()`
(*ironic.drivers.modules.agent.AgentRAID* method), 3064

- method*), 2533
- get_properties()
(*ironic.drivers.modules.agent.AgentRescue*
method), 2535
- get_properties()
(*ironic.drivers.modules.agent.CustomAgentDeploy*
method), 2544
- get_properties()
(*ironic.drivers.modules.agent_power.AgentPower*
method), 2627
- get_properties()
(*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
method), 1936
- get_properties()
(*ironic.drivers.modules.drac.bios.DracWSManBIOS*
method), 1949
- get_properties()
(*ironic.drivers.modules.drac.inspect.DracWSManInspect*
method), 1978
- get_properties()
(*ironic.drivers.modules.drac.management.DracWSManManagement*
method), 1998
- get_properties()
(*ironic.drivers.modules.drac.power.DracWSManPower*
method), 2011
- get_properties()
(*ironic.drivers.modules.drac.raid.DracWSManRAID*
method), 2030
- get_properties()
(*ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthru*
method), 2058
- get_properties()
(*ironic.drivers.modules.fake.FakeBIOS*
method), 2702
- get_properties()
(*ironic.drivers.modules.fake.FakeBoot*
method), 2707
- get_properties()
(*ironic.drivers.modules.fake.FakeConsole*
method), 2714
- get_properties()
(*ironic.drivers.modules.fake.FakeDeploy*
method), 2721
- get_properties()
(*ironic.drivers.modules.fake.FakeInspect*
method), 2729
- get_properties()
(*ironic.drivers.modules.fake.FakeManagement*
method), 2738
- get_properties()
(*ironic.drivers.modules.fake.FakePower*
method), 2759
- get_properties()
(*ironic.drivers.modules.fake.FakeRAID*
method), 2770
- get_properties()
(*ironic.drivers.modules.fake.FakeRescue*
method), 2771
- get_properties()
(*ironic.drivers.modules.fake.FakeStorage*
method), 2778
- get_properties()
(*ironic.drivers.modules.fake.FakeVendorA*
method), 2782
- get_properties()
(*ironic.drivers.modules.fake.FakeVendorB*
method), 2785
- get_properties()
(*ironic.drivers.modules.ibmcm.management.IBMCMManagement*
method), 2068
- get_properties()
(*ironic.drivers.modules.ibmcm.power.IBMCPower*
method), 2082
- get_properties()
(*ironic.drivers.modules.ibmcm.raid.IbmcRAID*
method), 2096
- get_properties()
(*ironic.drivers.modules.ibmcm.vendor.IBMCVendor*
method), 2101
- get_properties()
(*ironic.drivers.modules.ilo.bios.IloBIOS*
method), 2110
- get_properties()
(*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
method), 2119
- get_properties()
(*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*
method), 2131
- get_properties()
(*ironic.drivers.modules.ilo.console.IloConsoleInterface*
method), 2193
- get_properties()
(*ironic.drivers.modules.ilo.inspect.IloInspect*
method), 2200
- get_properties()
(*ironic.drivers.modules.ilo.management.IloManagement*
method), 2215
- get_properties()
(*ironic.drivers.modules.ilo.power.IloPower*
method), 2241
- get_properties()
(*ironic.drivers.modules.ilo.raid.Ilo5RAID*
method), 2241

- method), 2254
- get_properties() (ironic.drivers.modules.ilo.vendor.VendorPassthru method), 2257
- get_properties() (ironic.drivers.modules.inspector.Inspector method), 2814
- get_properties() (ironic.drivers.modules.ipmitool.IPMIConsole method), 2818
- get_properties() (ironic.drivers.modules.ipmitool.IPMIManagement method), 2823
- get_properties() (ironic.drivers.modules.ipmitool.IPMIPower method), 2834
- get_properties() (ironic.drivers.modules.ipmitool.VendorPassthru method), 2850
- get_properties() (ironic.drivers.modules.irmc.bios.IRMCBIOS method), 2264
- get_properties() (ironic.drivers.modules.irmc.boot.IRMCVirtualMedia method), 2272
- get_properties() (ironic.drivers.modules.irmc.inspect.IRMCInspect method), 2292
- get_properties() (ironic.drivers.modules.irmc.management.IRMCManagement method), 2297
- get_properties() (ironic.drivers.modules.irmc.power.IRMCPower method), 2318
- get_properties() (ironic.drivers.modules.irmc.raid.IRMCRAID method), 2329
- get_properties() (ironic.drivers.modules.noop.FailMixin method), 2863
- get_properties() (ironic.drivers.modules.noop_mgmt.NoopManagement method), 2890
- get_properties() (ironic.drivers.modules.pxe.PXEAnacondaDeploy method), 2903
- get_properties() (ironic.drivers.modules.pxe.PXERamdiskDeploy method), 2909
- get_properties() (ironic.drivers.modules.pxe_base.PXEBaseMixin method), 2916
- get_properties() (ironic.drivers.modules.redfish.bios.RedfishBIOS method), 2383
- get_properties() (ironic.drivers.modules.redfish.boot.RedfishVirtualMedia method), 2393
- get_properties() (ironic.drivers.modules.redfish.inspect.RedfishInspect method), 2403
- get_properties() (ironic.drivers.modules.redfish.management.RedfishManagement method), 2418
- get_properties() (ironic.drivers.modules.redfish.power.RedfishPower method), 2450
- get_properties() (ironic.drivers.modules.redfish.raid.RedfishRAID method), 2464
- get_properties() (ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru method), 2488
- get_properties() (ironic.drivers.modules.snmp.SNMPPower method), 2944
- get_properties() (ironic.drivers.modules.storage.cinder.CinderStorage method), 2493
- get_properties() (ironic.drivers.modules.storage.external.ExternalStorage method), 2498
- get_properties() (ironic.drivers.modules.storage.noop.NoopStorage method), 2503
- get_properties() (ironic.drivers.modules.xclarity.management.XClarityManagement method), 2514
- get_properties() (ironic.drivers.modules.xclarity.power.XClarityPower method), 2519
- get_properties() (ironic.drivers.utils.MixinVendorInterface method), 3138
- get_provisioning_network_uuid() (ironic.common.neutron.NeutronNetworkInterfaceMixin method), 1233
- get_pxe_boot_file() (in module ironic.drivers.modules.deploy_utils), 2668
- get_pxe_config_file_path() (in module ironic.common.pxe_utils), 1290

get_pxe_config_template() (in module *ironic.drivers.modules.deploy_utils*), 2669
 get_raid_controller_list() (in module *ironic.drivers.modules.ibm.vendor.IBMVendor* method), 2102
 get_raid_logical_disk_properties() (in module *ironic.conductor.manager.ConductorManager* method), 1401
 get_raid_logical_disk_properties() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 1484
 get_ramdisk_logs_file_name() (in module *ironic.drivers.utils*), 3148
 get_random_topic() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 1487
 get_remote_boot_volume() (in module *ironic.drivers.modules.deploy_utils*), 2670
 get_request_return_fields() (in module *ironic.api.controllers.v1.utils*), 987
 get_rescuing_network_uuid() (in module *ironic.common.neutron.NeutronNetworkInterface* method), 1233
 get_ring() (in module *ironic.common.hash_ring.HashRingManager* method), 1183
 get_root_device_for_deploy() (in module *ironic.drivers.modules.deploy_utils*), 2671
 get_root_dir() (in module *ironic.common.pxe_utils*), 1291
 get_rpc_allocation() (in module *ironic.api.controllers.v1.utils*), 990
 get_rpc_allocation_with_suffix() (in module *ironic.api.controllers.v1.utils*), 991
 get_rpc_deploy_template() (in module *ironic.api.controllers.v1.utils*), 993
 get_rpc_deploy_template_with_suffix() (in module *ironic.api.controllers.v1.utils*), 995
 get_rpc_node() (in module *ironic.api.controllers.v1.utils*), 996
 get_rpc_node_with_suffix() (in module *ironic.api.controllers.v1.utils*), 998
 get_rpc_portgroup() (in module *ironic.api.controllers.v1.utils*), 999
 get_rpc_portgroup_with_suffix() (in module *ironic.api.controllers.v1.utils*), 1000
 get_secure_boot_mode() (in module *ironic.drivers.modules.ilo.common*), 2166
 get_secure_boot_mode() (in module *ironic.drivers.modules.irmc.common*), 2287
 get_secure_boot_state() (in module *ironic.drivers.base.ManagementInterface* method), 3013
 get_secure_boot_state() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 2215
 get_secure_boot_state() (in module *ironic.drivers.modules.irmc.management.IRMCMangement* method), 2297
 get_secure_boot_state() (in module *ironic.drivers.modules.redfish.management.RedfishManag* method), 2419
 get_sensors_data() (in module *ironic.drivers.base.ManagementInterface* method), 3015
 get_sensors_data() (in module *ironic.drivers.modules.drac.management.DracWSManM* method), 1998
 get_sensors_data() (in module *ironic.drivers.modules.fake.FakeManagement* method), 2738
 get_sensors_data() (in module *ironic.drivers.modules.ibm.management.IBMCMangement* method), 2068
 get_sensors_data() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 2217
 get_sensors_data() (in module *ironic.drivers.modules.ipmitool.IPMIMangement* method), 2824
 get_sensors_data() (in module *ironic.drivers.modules.irmc.management.IRMCMangement* method), 2299
 get_sensors_data() (in module *ironic.drivers.modules.noop_mgmt.NoopManagement* method), 2890
 get_sensors_data() (in module *ironic.drivers.modules.redfish.management.RedfishManag* method), 2420
 get_sensors_data() (in module *ironic.drivers.modules.xclarity.management.XClarityMan* method), 2514
 get_sensors_notifier() (in module *ironic.common.rpc*), 1313
 get_server() (in module *ironic.common.rpc*), 1313

- `get_server_hardware_id()` (in module `ironic.drivers.modules.xclarity.common`), 2507
- `get_server_post_state()` (in module `ironic.drivers.modules.ilo.common`), 2168
- `get_service_auth()` (in module `ironic.common.keystone`), 1220
- `get_session()` (in module `ironic.common.keystone`), 1221
- `get_shellinabox_console_url()` (in module `ironic.drivers.modules.console_utils`), 2643
- `get_single_nic_with_vif_port_id()` (in module `ironic.drivers.modules.deploy_utils`), 2672
- `get_socat_console_url()` (in module `ironic.drivers.modules.console_utils`), 2644
- `get_source_format()` (in module `ironic.common.images`), 1213
- `get_steps()` (in module `ironic.drivers.modules.agent_base`), 2582
- `get_supported_boot_devices()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_supported_boot_devices()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1487
- `get_supported_boot_devices()` (`ironic.drivers.base.ManagementInterface` method), 3021
- `get_supported_boot_devices()` (`ironic.drivers.modules.drac.management.DracManagement` method), 2000
- `get_supported_boot_devices()` (`ironic.drivers.modules.fake.FakeManagement` method), 2744
- `get_supported_boot_devices()` (`ironic.drivers.modules.ibmcm.management.IBMCMManagement` method), 2069
- `get_supported_boot_devices()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 2219
- `get_supported_boot_devices()` (`ironic.drivers.modules.ipmitool.IPMIManagement` method), 2826
- `get_supported_boot_devices()` (`ironic.drivers.modules.noop_mgmt.NoopManagement` method), 2896
- `get_supported_boot_devices()` (`ironic.drivers.modules.redfish.management.RedfishManagement` method), 2422
- `get_supported_boot_devices()` (`ironic.drivers.modules.xclarity.management.XClarityManagement` method), 2515
- `get_supported_boot_modes()` (`ironic.drivers.base.ManagementInterface` method), 3021
- `get_supported_boot_modes()` (`ironic.drivers.modules.ibmcm.management.IBMCMManagement` method), 2071
- `get_supported_boot_modes()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 2220
- `get_supported_boot_modes()` (`ironic.drivers.modules.redfish.management.RedfishManagement` method), 2423
- `get_supported_indicators()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `get_supported_indicators()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1490
- `get_supported_indicators()` (`ironic.drivers.base.ManagementInterface` method), 3024
- `get_supported_indicators()` (`ironic.drivers.modules.fake.FakeManagement` method), 2745
- `get_supported_indicators()` (`ironic.drivers.modules.redfish.management.RedfishManagement` method), 2424
- `get_supported_power_states()` (`ironic.drivers.base.PowerInterface` method), 3065
- `get_supported_power_states()` (`ironic.drivers.modules.agent_power.AgentPower` method), 2628
- `get_supported_power_states()` (`ironic.drivers.modules.fake.FakePower` method), 2759
- `get_supported_power_states()` (`ironic.drivers.modules.ibmcm.power.IBMCPower` method), 2082
- `get_supported_power_states()` (`ironic.drivers.modules.ilo.power.IloPower` method), 2241
- `get_supported_power_states()` (`ironic.drivers.modules.ipmitool.IPMIPower` method), 2826

method), 2834

get_supported_power_states() (in module *ironic.drivers.modules.irmc.power.IRMCPower* method), 2318

get_supported_power_states() (in module *ironic.drivers.modules.redfish.power.RedfishPower* method), 2451

get_swift_session() (in module *ironic.common.swift*), 1342

get_swift_url() (in module *ironic.drivers.modules.ilo.firmware_processor*), 2199

get_system() (in module *ironic.drivers.modules.redfish.utils*), 2481

get_target_version() (in module *ironic.objects.base.IronicObject* class method), 3175

get_task_monitor() (in module *ironic.drivers.modules.redfish.utils*), 2482

get_temp_url() (in module *ironic.common.swift.SwiftAPI* method), 1338

get_temp_url_for_glance_image() (in module *ironic.common.images*), 1213

get_tftp_path_prefix() (in module *ironic.common.pxe_utils*), 1291

get_topic_for() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 1493

get_topic_for_driver() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 1494

get_trait_names() (in module *ironic.objects.trait.TraitList* method), 3366

get_transport_url() (in module *ironic.common.rpc*), 1313

get_update_service() (in module *ironic.drivers.modules.redfish.utils*), 2483

get_updated_capabilities() (in module *ironic.common.utils*), 1347

get_vendor_passthru_metadata() (in module *ironic.conductor.manager*), 1410

get_versioned_notifier() (in module *ironic.common.rpc*), 1313

get_volume_connector_by_id() (in module *ironic.db.api.Connection* method), 1855

get_volume_connector_by_id() (in module *ironic.db.sqlalchemy.api.Connection* method), 1702

get_volume_connector_by_uuid() (in module *ironic.db.api.Connection* method), 1857

get_volume_connector_by_uuid() (in module *ironic.db.sqlalchemy.api.Connection* method), 1704

get_volume_connector_list() (in module *ironic.db.api.Connection* method), 1858

get_volume_connector_list() (in module *ironic.db.sqlalchemy.api.Connection* method), 1705

get_volume_connectors_by_node_id() (in module *ironic.db.api.Connection* method), 1860

get_volume_connectors_by_node_id() (in module *ironic.db.sqlalchemy.api.Connection* method), 1707

get_volume_pxe_options() (in module *ironic.common.pxe_utils*), 1292

get_volume_target_by_id() (in module *ironic.db.api.Connection* method), 1863

get_volume_target_by_id() (in module *ironic.db.sqlalchemy.api.Connection* method), 1710

get_volume_target_by_uuid() (in module *ironic.db.api.Connection* method), 1864

get_volume_target_by_uuid() (in module *ironic.db.sqlalchemy.api.Connection* method), 1711

get_volume_target_list() (in module *ironic.db.api.Connection* method), 1865

get_volume_target_list() (in module *ironic.db.sqlalchemy.api.Connection* method), 1712

get_volume_targets_by_node_id() (in module *ironic.db.api.Connection* method), 1867

get_volume_targets_by_node_id() (in module *ironic.db.sqlalchemy.api.Connection* method), 1714

get_volume_targets_by_volume_id() (in module *ironic.db.api.Connection* method), 1870

get_volume_targets_by_volume_id() (in module *ironic.db.sqlalchemy.api.Connection* method), 1717

get_xclarity_client() (in module

- `ironic.drivers.modules.xclarity.common`), 2508
- `getargspec()` (in module `ironic.api.functions`), 1068
- `GlanceConnectionFailed`, 1159
- `GlanceImageService` (class in `ironic.common.glance_service.image_service`), 1084
- ## H
- `handle_error_notification()` (in module `ironic.api.controllers.v1.notification_utils`), 903
- `handle_ibmc_exception()` (in module `ironic.drivers.modules.ibmc.utils`), 2097
- `handle_signal()` (`ironic.common.rpc_service.RPCService` method), 1313
- `handle_sync_power_state_max_retries_exception()` (in module `ironic.conductor.manager`), 1410
- `hardware_type` (`ironic.db.sqlalchemy.models.ConductorHardware` attribute), 1769
- `hardware_types()` (in module `ironic.common.driver_factory`), 1151
- `HardwareInspectionFailure`, 1159
- `HardwareTypesFactory` (class in `ironic.common.driver_factory`), 1137
- `has_next()` (in module `ironic.api.controllers.v1.collection`), 819
- `hash_password()` (in module `ironic.conductor.utils`), 1585
- `HashRingManager` (class in `ironic.common.hash_ring`), 1183
- `head_object()` (`ironic.common.swift.SwiftAPI` method), 1339
- `heartbeat()` (`ironic.conductor.manager.ConductorManager` method), 1401
- `heartbeat()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1495
- `heartbeat()` (`ironic.drivers.base.DeployInterface` method), 2990
- `heartbeat()` (`ironic.drivers.modules.agent_base.HeartbeatMixin` method), 2577
- `heartbeat_allowed_states` (`ironic.drivers.modules.agent_base.HeartbeatMixin` property), 2578
- `HeartbeatController` (class in `ironic.api.controllers.v1.ramdisk`), 924
- `HeartbeatMixin` (class in `ironic.drivers.modules.agent_base`), 2576
- `hide_fields_in_newer_versions()` (in module `ironic.api.controllers.v1.allocation`), 811
- `hide_fields_in_newer_versions()` (in module `ironic.api.controllers.v1.driver`), 840
- `hide_fields_in_newer_versions()` (in module `ironic.api.controllers.v1.node`), 897
- `hide_fields_in_newer_versions()` (in module `ironic.api.controllers.v1.port`), 915
- `host` (`ironic.objects.notification.NotificationPublisher` property), 3317
- `hostname` (`ironic.db.sqlalchemy.models.Conductor` attribute), 1769
- `hostname` (`ironic.objects.conductor.Conductor` property), 3210
- `HTTPForbidden`, 1159
- `HttpImageService` (class in `ironic.common.image_service`), 1190
- ## HTTP Found Interfaces
- ## I
- `IBMConnectionError`, 1159
- `IBMError`, 1159
- `IBMCHardware` (class in `ironic.drivers.ibmc`), 3125
- `IBMManagement` (class in `ironic.drivers.modules.ibmc.management`), 2063
- `IBMCPower` (class in `ironic.drivers.modules.ibmc.power`), 2079
- `IbmCRAID` (class in `ironic.drivers.modules.ibmc.raid`), 2089
- `IBMVendor` (class in `ironic.drivers.modules.ibmc.vendor`), 2099
- `id` (`ironic.db.sqlalchemy.models.Allocation` attribute), 1766
- `id` (`ironic.db.sqlalchemy.models.Chassis` attribute), 1768
- `id` (`ironic.db.sqlalchemy.models.Conductor` attribute), 1769
- `id` (`ironic.db.sqlalchemy.models.ConductorHardwareInterfaces` attribute), 1769
- `id` (`ironic.db.sqlalchemy.models.DeployTemplate` attribute), 1770

- `id` (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 1771
- `id` (*ironic.db.sqlalchemy.models.Node* attribute), 1773
- `id` (*ironic.db.sqlalchemy.models.Port* attribute), 1777
- `id` (*ironic.db.sqlalchemy.models.Portgroup* attribute), 1778
- `id` (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 1779
- `id` (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780
- `id` (*ironic.objects.allocation.Allocation* property), 3160
- `id` (*ironic.objects.chassis.Chassis* property), 3202
- `id` (*ironic.objects.conductor.Conductor* property), 3210
- `id` (*ironic.objects.deploy_template.DeployTemplate* property), 3224
- `id` (*ironic.objects.node.Node* property), 3272
- `id` (*ironic.objects.port.Port* property), 3326
- `id` (*ironic.objects.portgroup.Portgroup* property), 3347
- `id` (*ironic.objects.volume_connector.VolumeConnector* property), 3375
- `id` (*ironic.objects.volume_target.VolumeTarget* property), 3393
- `IDRACHardware` (class in *ironic.drivers.drac*), 3110
- `ignore_extra_args` (*ironic.api.functions.FunctionDefinition* attribute), 1067
- `Ilo5Hardware` (class in *ironic.drivers.ilo*), 3126
- `Ilo5Management` (class in *ironic.drivers.modules.ilo.management*), 2204
- `Ilo5RAID` (class in *ironic.drivers.modules.ilo.raid*), 2247
- `IloBIOS` (class in *ironic.drivers.modules.ilo.bios*), 2106
- `IloConsoleInterface` (class in *ironic.drivers.modules.ilo.console*), 2192
- `IloHardware` (class in *ironic.drivers.ilo*), 3127
- `IloInspect` (class in *ironic.drivers.modules.ilo.inspect*), 2200
- `IloiPXEBoot` (class in *ironic.drivers.modules.ilo.boot*), 2140
- `IloManagement` (class in *ironic.drivers.modules.ilo.management*), 2207
- `IloOperationError`, 1160
- `IloOperationNotSupported`, 1160
- `IloPower` (class in *ironic.drivers.modules.ilo.power*), 2239
- `IloPXEBoot` (class in *ironic.drivers.modules.ilo.boot*), 2112
- `IloUefiHttpsBoot` (class in *ironic.drivers.modules.ilo.boot*), 2117
- `IloVirtualMediaBoot` (class in *ironic.drivers.modules.ilo.boot*), 2128
- `image_checksum` (*ironic.objects.deployment.Deployment* property), 3238
- `image_ref` (*ironic.objects.deployment.Deployment* property), 3238
- `image_show()` (in *ironic.common.images*), 1214
- `image_to_raw()` (in *ironic.common.images*), 1214
- `ImageCache` (class in *ironic.drivers.modules.image_cache*), 2787
- `ImageConvertFailed`, 1160
- `ImageCreationFailed`, 1160
- `ImageDownloadFailed`, 1160
- `ImageHandler` (class in *ironic.drivers.modules.image_utils*), 2793
- `ImageNotAuthorized`, 1160
- `ImageNotFound`, 1161
- `ImageRefValidationFailed`, 1161
- `ImageUnacceptable`, 1161
- `ImageUploadFailed`, 1161
- `import_configuration()` (*ironic.drivers.modules.drac.management.DracRedfishManagement* method), 1990
- `IMPORT_CONFIGURATION_ARGSINFO` (*ironic.drivers.modules.drac.management.DracRedfishManagement* attribute), 1987
- `import_export_configuration()` (*ironic.drivers.modules.drac.management.DracRedfishManagement* method), 1992
- `IMPORT_EXPORT_CONFIGURATION_ARGSINFO` (*ironic.drivers.modules.drac.management.DracRedfishManagement* attribute), 1987
- `in_core_deploy_step()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeployment* method), 1936
- `IncompatibleInterface`, 1161
- `IncompleteLookup`, 1161

index() (*ironic.api.controllers.root.RootController* method), 1056
 index() (*ironic.api.controllers.v1.Controller* method), 1054
 indicator_convert_with_links() (in module *ironic.api.controllers.v1.node*), 897
 indicator_list_from_dict() (in module *ironic.api.controllers.v1.node*), 898
 IndicatorAtComponent (class in *ironic.api.controllers.v1.node*), 846
 IndicatorController (class in *ironic.api.controllers.v1.node*), 847
 indicators (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 856
 INFO (*ironic.objects.fields.NotificationLevel* attribute), 3247
 init() (in module *ironic.common.rpc*), 1313
 init_enforcer() (in module *ironic.common.policy*), 1264
 init_host() (*ironic.conductor.base_manager.BaseConductorManager* method), 1378
 initial_node_provision_state() (in module *ironic.api.controllers.v1.utils*), 1002
 initialize() (*ironic.common.fsm.FSM* method), 1180
 initialize_wsgi_app() (in module *ironic.api.wsgi*), 1080
 inject_nmi (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 856
 inject_nmi() (*ironic.conductor.manager.ConductorManager* method), 1401
 inject_nmi() (*ironic.conductor.rpcapi.ConductorAPI* method), 1498
 inject_nmi() (*ironic.drivers.base.ManagementInterface* method), 3032
 inject_nmi() (*ironic.drivers.modules.ibmcm.manager.IBMCMManagementInterface* method), 2072
 inject_nmi() (*ironic.drivers.modules.ilo.management.IloManagementInterface* method), 2221
 inject_nmi() (*ironic.drivers.modules.ipmitool.IPMIManagementInterface* method), 2826
 inject_nmi() (*ironic.drivers.modules.irmc.management.IRMCManagementInterface* method), 2307
 inject_nmi() (*ironic.drivers.modules.redfish.management.RedfishManagementInterface* method), 2432
 InjectNmiController (class in *ironic.api.controllers.v1.node*), 850
 InputFileError, 1161
 inspect (*ironic.drivers.base.BareDriver* attribute), 2960
 inspect_hardware() (*ironic.conductor.manager.ConductorManager* method), 1401
 inspect_hardware() (*ironic.conductor.rpcapi.ConductorAPI* method), 1500
 inspect_hardware() (*ironic.drivers.base.InspectInterface* method), 3002
 inspect_hardware() (*ironic.drivers.modules.drac.inspect.DracRedfishInspect* method), 1976
 inspect_hardware() (*ironic.drivers.modules.drac.inspect.DracWSManInspect* method), 1978
 inspect_hardware() (*ironic.drivers.modules.fake.FakeInspect* method), 2729
 inspect_hardware() (*ironic.drivers.modules.ilo.inspect.IloInspect* method), 2201
 inspect_hardware() (*ironic.drivers.modules.inspector.Inspector* method), 2814
 inspect_hardware() (*ironic.drivers.modules.irmc.inspect.IRMCInspect* method), 2292
 inspect_hardware() (*ironic.drivers.modules.noop.NoInspect* method), 2874
 inspect_hardware() (*ironic.drivers.modules.redfish.inspect.RedfishInspect* method), 2404
 inspect_interface (*ironic.db.sqlalchemy.models.Node* attribute), 1773
 inspect_ibmcm_management (*ironic.objects.node.Node* property), 3272
 inspect_ilo_management (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
 inspect_ipmi_management (*ironic.objects.node.NodeCRUDPayload* property), 3286
 inspect_redfish_management (*ironic.objects.node.NodePayload* property), 3299
 inspect_interface (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
 inspect_interface (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3304

property), 3310

INSPECTFAIL (in module *ironic.common.states*), 1324

INSPECTING (in module *ironic.common.states*), 1325

InspectInterface (class in *ironic.drivers.base*), 2999

inspection_finished_at (ironic.db.sqlalchemy.models.Node attribute), 1773

inspection_finished_at (ironic.objects.node.Node property), 3272

inspection_finished_at (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293

inspection_finished_at (ironic.objects.node.NodeCRUDPayload property), 3286

inspection_finished_at (ironic.objects.node.NodePayload property), 3299

inspection_finished_at (ironic.objects.node.NodeSetPowerStatePayload property), 3304

inspection_finished_at (ironic.objects.node.NodeSetProvisionStatePayload property), 3310

inspection_started_at (ironic.db.sqlalchemy.models.Node attribute), 1773

inspection_started_at (ironic.objects.node.Node property), 3272

inspection_started_at (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293

inspection_started_at (ironic.objects.node.NodeCRUDPayload property), 3286

inspection_started_at (ironic.objects.node.NodePayload property), 3299

inspection_started_at (ironic.objects.node.NodeSetPowerStatePayload property), 3304

inspection_started_at (ironic.objects.node.NodeSetProvisionStatePayload property), 3310

Inspector (class in *ironic.drivers.modules.inspector*), 2813

INSPECTWAIT (in module *ironic.common.states*), 1325

install_bootloader() (ironic.drivers.modules.agent_client.AgentClient method), 2615

instance_info (ironic.db.sqlalchemy.models.Node attribute), 1773

instance_info (ironic.objects.node.Node property), 3272

instance_info (ironic.objects.node.NodeCRUDPayload property), 3286

instance_info (ironic.objects.node.NodeSetProvisionStatePayload property), 3310

instance_info_mapping (ironic.objects.deployment.Deployment attribute), 3238

instance_info_mapping_rev (ironic.objects.deployment.Deployment attribute), 3238

instance_uuid (ironic.db.sqlalchemy.models.Node attribute), 1773

instance_uuid (ironic.objects.node.Node property), 3272

instance_uuid (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293

instance_uuid (ironic.objects.node.NodeCRUDPayload property), 3286

instance_uuid (ironic.objects.node.NodePayload property), 3299

instance_uuid (ironic.objects.node.NodeSetPowerStatePayload property), 3304

instance_uuid (ironic.objects.node.NodeSetProvisionStatePayload property), 3310

InstanceAssociated, 1162

InstanceDeployFailure, 1162

InstanceImageCache (class in *ironic.drivers.modules.deploy_utils*), 2650

InstanceNotFound, 1162

InstanceRescueFailure, 1162

InstanceUnrescueFailure, 1162

InsufficientMemory, 1162

InsufficientDiskSpace, 1162

integer() (in module *ironic.common.args*), 1097

IntegerField (class in *ironic.objects.fields*), 3244

IntelIPMIHardware (class in *ironic.drivers.intel_ipmi*), 3130

IntelIPMIManagement (class in *ironic.drivers.modules.intel_ipmi.management*), 2260

interface (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 1777
 attribute), 1771
 interface_name
 (*ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 1770
 interface_type
 (*ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 1770
 interface_type
 (*ironic.drivers.base.BaseInterface* attribute), 2973
 interface_type
 (*ironic.drivers.base.BIOSInterface* attribute), 2957
 interface_type
 (*ironic.drivers.base.BootInterface* attribute), 2978
 interface_type
 (*ironic.drivers.base.ConsoleInterface* attribute), 2985
 interface_type
 (*ironic.drivers.base.DeployInterface* attribute), 2991
 interface_type
 (*ironic.drivers.base.InspectInterface* attribute), 3003
 interface_type
 (*ironic.drivers.base.ManagementInterface* attribute), 3033
 interface_type
 (*ironic.drivers.base.NetworkInterface* attribute), 3051
 interface_type
 (*ironic.drivers.base.PowerInterface* attribute), 3066
 interface_type
 (*ironic.drivers.base.RAIDInterface* attribute), 3079
 interface_type
 (*ironic.drivers.base.RescueInterface* attribute), 3085
 interface_type
 (*ironic.drivers.base.StorageInterface* attribute), 3089
 interface_type
 (*ironic.drivers.base.VendorInterface* attribute), 3093
 InterfaceNotFoundInEntrypoint, 1162
 interfaces() (in *module ironic.common.driver_factory*), 1151
 internal_info (*ironic.db.sqlalchemy.models.Port* attribute), 1778
 internal_info (*ironic.objects.port.Port* property), 3326
 internal_info (*ironic.objects.portgroup.Portgroup* property), 3347
 Invalid, 1163
 invalid_sort_key_list
 (*ironic.api.controllers.v1.allocation.AllocationsController* attribute), 809
 invalid_sort_key_list
 (*ironic.api.controllers.v1.allocation.NodeAllocationController* attribute), 811
 invalid_sort_key_list
 (*ironic.api.controllers.v1.chassis.ChassisController* attribute), 817
 invalid_sort_key_list
 (*ironic.api.controllers.v1.conductor.CollectorsController* attribute), 825
 invalid_sort_key_list
 (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* attribute), 830
 invalid_sort_key_list
 (*ironic.api.controllers.v1.node.NodesController* attribute), 893
 invalid_sort_key_list
 (*ironic.api.controllers.v1.port.PortsController* attribute), 913
 invalid_sort_key_list
 (*ironic.api.controllers.v1.portgroup.PortgroupsController* attribute), 922
 invalid_sort_key_list
 (*ironic.api.controllers.v1.volume_connector.VolumeConnectorController* attribute), 1032
 invalid_sort_key_list
 (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* attribute), 1047
 InvalidConductorGroup, 1163
 InvalidDatapathID, 1163
 InvalidDeployTemplate, 1163
 InvalidEndpoint, 1163
 InvalidIdentity, 1164
 InvalidImageRef, 1164
 InvalidInput, 1164
 InvalidIPAddress, 1163
 InvalidIPv4Address, 1163
 InvalidKickstartFile, 1164
 InvalidKickstartTemplate, 1164
 InvalidMAC, 1164
 InvalidName, 1164

InvalidParameterValue, 1165		ironic.api
InvalidState, 1165		module, 1081
InvalidStateRequested, 1165		ironic.api.app
InvalidSwitchID, 1165		module, 1063
InvalidUUID, 1165		ironic.api.config
InvalidUuidOrName, 1165		module, 1064
IPMIConsole	(class in <i>ironic.drivers.modules.ipmitool</i>), 2817	ironic.api.controllers
		module, 1059
IPMIFailure, 1159		ironic.api.controllers.base
		module, 1054
IPMIHardware	(class in <i>ironic.drivers.ipmi</i>), 3131	
IPMIManagement	(class in <i>ironic.drivers.modules.ipmitool</i>), 2819	ironic.api.controllers.link
		module, 1056
IPMIPower	(class in <i>ironic.drivers.modules.ipmitool</i>), 2832	ironic.api.controllers.root
		module, 1056
IPMIShellinaboxConsole	(class in <i>ironic.drivers.modules.ipmitool</i>), 2841	ironic.api.controllers.v1
		module, 1053
IPMISocatConsole	(class in <i>ironic.drivers.modules.ipmitool</i>), 2844	ironic.api.controllers.v1.allocation
		module, 805
ipxe_enabled	(<i>ironic.drivers.modules.ipxe.iPXEBoot</i> attribute), 2862	ironic.api.controllers.v1.bios
		module, 811
ipxe_enabled	(<i>ironic.drivers.modules.pxe_base.PXEBoot</i> attribute), 2917	ironic.api.controllers.v1.chassis
		module, 813
iPXEBoot	(class in <i>ironic.drivers.modules.ipxe</i>), 2862	ironic.api.controllers.v1.collection
		module, 819
IRMCBIOS	(class in <i>ironic.drivers.modules.irmc.bios</i>), 2261	ironic.api.controllers.v1.conductor
		module, 822
IRMCHardware	(class in <i>ironic.drivers.irmc</i>), 3133	ironic.api.controllers.v1.deploy_template
IRMCInspect	(class in <i>ironic.drivers.modules.irmc.inspect</i>), 2291	module, 826
		ironic.api.controllers.v1.driver
		module, 833
IRMCManagement	(class in <i>ironic.drivers.modules.irmc.management</i>), 2297	ironic.api.controllers.v1.event
		module, 842
IRMCOperationError, 1160		ironic.api.controllers.v1.node
		module, 843
IRMCPower	(class in <i>ironic.drivers.modules.irmc.power</i>), 2316	ironic.api.controllers.v1.notification_utils
		module, 901
IRMC PXEBoot	(class in <i>ironic.drivers.modules.irmc.boot</i>), 2267	ironic.api.controllers.v1.port
		module, 905
		ironic.api.controllers.v1.portgroup
		module, 916
IRMCRAID	(class in <i>ironic.drivers.modules.irmc.raid</i>), 2325	ironic.api.controllers.v1.ramdisk
		module, 924
IRMCSharedFileSystemNotMounted, 1160		ironic.api.controllers.v1.utils
IRMCVirtualMediaBoot	(class in <i>ironic.drivers.modules.irmc.boot</i>), 2269	module, 931
		ironic.api.controllers.v1.versions
		module, 1022
IRMCVolumeBootMixIn	(class in <i>ironic.drivers.modules.irmc.boot</i>), 2280	ironic.api.controllers.v1.volume
		module, 1023
		ironic.api.controllers.v1.volume_connector
		module, 1023
ironic		ironic.api.controllers.v1.volume_target
	module, 3407	

- module, 1038
- ironic.api.controllers.version
 - module, 1057
- ironic.api.functions
 - module, 1064
- ironic.api.hooks
 - module, 1071
- ironic.api.method
 - module, 1079
- ironic.api.middleware
 - module, 1061
- ironic.api.middleware.auth_public_routes
 - module, 1059
- ironic.api.middleware.json_ext
 - module, 1060
- ironic.api.middleware.parsable_error
 - module, 1060
- ironic.api.wsgi
 - module, 1080
- ironic.cmd
 - module, 1084
- ironic.cmd.api
 - module, 1081
- ironic.cmd.conductor
 - module, 1081
- ironic.cmd.dbsync
 - module, 1081
- ironic.cmd.status
 - module, 1083
- ironic.common
 - module, 1372
- ironic.common.args
 - module, 1093
- ironic.common.boot_devices
 - module, 1110
- ironic.common.boot_modes
 - module, 1113
- ironic.common.cinder
 - module, 1114
- ironic.common.components
 - module, 1128
- ironic.common.config
 - module, 1130
- ironic.common.context
 - module, 1130
- ironic.common.dhcp_factory
 - module, 1132
- ironic.common.driver_factory
 - module, 1136
- ironic.common.exception
 - module, 1152
- ironic.common.faults
 - module, 1176
- ironic.common.fsm
 - module, 1177
- ironic.common.glance_service
 - module, 1093
- ironic.common.glance_service.image_service
 - module, 1084
- ironic.common.glance_service.service_utils
 - module, 1091
- ironic.common.hash_ring
 - module, 1183
- ironic.common.i18n
 - module, 1183
- ironic.common.image_service
 - module, 1184
- ironic.common.images
 - module, 1197
- ironic.common.indicator_states
 - module, 1216
- ironic.common.keystone
 - module, 1216
- ironic.common.molds
 - module, 1223
- ironic.common.network
 - module, 1226
- ironic.common.neutron
 - module, 1233
- ironic.common.nova
 - module, 1260
- ironic.common.policy
 - module, 1262
- ironic.common.profiler
 - module, 1266
- ironic.common.pxe_utils
 - module, 1269
- ironic.common.raid
 - module, 1300
- ironic.common.release_mappings
 - module, 1307
- ironic.common.rpc
 - module, 1309
- ironic.common.rpc_service
 - module, 1313
- ironic.common.service
 - module, 1315
- ironic.common.states
 - module, 1315
- ironic.common.swift
 - module, 1334
- ironic.common.utils

- module, 1342
- ironic.common.wsgi_service
 - module, 1370
- ironic.conductor
 - module, 1620
- ironic.conductor.allocations
 - module, 1372
- ironic.conductor.base_manager
 - module, 1377
- ironic.conductor.cleaning
 - module, 1383
- ironic.conductor.deployments
 - module, 1388
- ironic.conductor.manager
 - module, 1393
- ironic.conductor.notification_utils
 - module, 1412
- ironic.conductor.rpcapi
 - module, 1418
- ironic.conductor.steps
 - module, 1548
- ironic.conductor.task_manager
 - module, 1554
- ironic.conductor.utils
 - module, 1573
- ironic.conf
 - module, 1628
- ironic.conf.agent
 - module, 1620
- ironic.conf.anaconda
 - module, 1621
- ironic.conf.ansible
 - module, 1621
- ironic.conf.api
 - module, 1621
- ironic.conf.audit
 - module, 1621
- ironic.conf.auth
 - module, 1621
- ironic.conf.cinder
 - module, 1622
- ironic.conf.conductor
 - module, 1623
- ironic.conf.console
 - module, 1623
- ironic.conf.database
 - module, 1623
- ironic.conf.default
 - module, 1623
- ironic.conf.deploy
 - module, 1623
- ironic.conf.dhcp
 - module, 1623
- ironic.conf.drac
 - module, 1623
- ironic.conf.glance
 - module, 1623
- ironic.conf.healthcheck
 - module, 1624
- ironic.conf.ibmc
 - module, 1624
- ironic.conf.ilo
 - module, 1624
- ironic.conf.inspector
 - module, 1624
- ironic.conf.ipmi
 - module, 1624
- ironic.conf.irmc
 - module, 1624
- ironic.conf.metrics
 - module, 1624
- ironic.conf.metrics_statsd
 - module, 1624
- ironic.conf.molds
 - module, 1625
- ironic.conf.neutron
 - module, 1625
- ironic.conf.nova
 - module, 1625
- ironic.conf.opts
 - module, 1625
- ironic.conf.pxe
 - module, 1627
- ironic.conf.redfish
 - module, 1627
- ironic.conf.service_catalog
 - module, 1627
- ironic.conf.snmp
 - module, 1627
- ironic.conf.swift
 - module, 1627
- ironic.conf.xclarity
 - module, 1628
- ironic.db
 - module, 1911
- ironic.db.api
 - module, 1781
- ironic.db.migration
 - module, 1911
- ironic.db.sqlalchemy
 - module, 1781
- ironic.db.sqlalchemy.api

- module, 1628
- ironic.db.sqlalchemy.migration
 - module, 1762
- ironic.db.sqlalchemy.models
 - module, 1765
- ironic.dhcp
 - module, 1933
- ironic.dhcp.base
 - module, 1911
- ironic.dhcp.neutron
 - module, 1919
- ironic.dhcp.none
 - module, 1926
- ironic.drivers
 - module, 3153
- ironic.drivers.base
 - module, 2950
- ironic.drivers.drac
 - module, 3110
- ironic.drivers.fake_hardware
 - module, 3112
- ironic.drivers.generic
 - module, 3116
- ironic.drivers.hardware_type
 - module, 3120
- ironic.drivers.ibmc
 - module, 3125
- ironic.drivers.ilo
 - module, 3126
- ironic.drivers.intel_ipmi
 - module, 3130
- ironic.drivers.ipmi
 - module, 3131
- ironic.drivers.irmc
 - module, 3132
- ironic.drivers.modules
 - module, 2950
- ironic.drivers.modules.agent
 - module, 2523
- ironic.drivers.modules.agent_base
 - module, 2553
- ironic.drivers.modules.agent_client
 - module, 2590
- ironic.drivers.modules.agent_power
 - module, 2625
- ironic.drivers.modules.ansible
 - module, 1943
- ironic.drivers.modules.ansible.deploy
 - module, 1933
- ironic.drivers.modules.boot_mode_utils
 - module, 2634
- ironic.drivers.modules.console_utils
 - module, 2643
- ironic.drivers.modules.deploy_utils
 - module, 2650
- ironic.drivers.modules.drac
 - module, 2063
- ironic.drivers.modules.drac.bios
 - module, 1943
- ironic.drivers.modules.drac.boot
 - module, 1967
- ironic.drivers.modules.drac.common
 - module, 1972
- ironic.drivers.modules.drac.inspect
 - module, 1975
- ironic.drivers.modules.drac.job
 - module, 1981
- ironic.drivers.modules.drac.management
 - module, 1985
- ironic.drivers.modules.drac.power
 - module, 2007
- ironic.drivers.modules.drac.raid
 - module, 2016
- ironic.drivers.modules.drac.utils
 - module, 2048
- ironic.drivers.modules.drac.vendor_passthru
 - module, 2051
- ironic.drivers.modules.fake
 - module, 2695
- ironic.drivers.modules.ibmc
 - module, 2106
- ironic.drivers.modules.ibmc.management
 - module, 2063
- ironic.drivers.modules.ibmc.mappings
 - module, 2079
- ironic.drivers.modules.ibmc.power
 - module, 2079
- ironic.drivers.modules.ibmc.raid
 - module, 2089
- ironic.drivers.modules.ibmc.utils
 - module, 2097
- ironic.drivers.modules.ibmc.vendor
 - module, 2099
- ironic.drivers.modules.ilo
 - module, 2259
- ironic.drivers.modules.ilo.bios
 - module, 2106
- ironic.drivers.modules.ilo.boot
 - module, 2112
- ironic.drivers.modules.ilo.common
 - module, 2149
- ironic.drivers.modules.ilo.console

- module, 2192
- ironic.drivers.modules.ilo.firmware_processormodule, 2358
 - module, 2194
- ironic.drivers.modules.ilo.inspect
 - module, 2200
- ironic.drivers.modules.ilo.management
 - module, 2204
- ironic.drivers.modules.ilo.power
 - module, 2239
- ironic.drivers.modules.ilo.raid
 - module, 2247
- ironic.drivers.modules.ilo.vendor
 - module, 2254
- ironic.drivers.modules.image_cache
 - module, 2787
- ironic.drivers.modules.image_utils
 - module, 2793
- ironic.drivers.modules.inspect_utils
 - module, 2811
- ironic.drivers.modules.inspector
 - module, 2813
- ironic.drivers.modules.intel_ipmi
 - module, 2260
- ironic.drivers.modules.intel_ipmi.managementmodule, 2260
 - module, 2260
- ironic.drivers.modules.ipmitool
 - module, 2816
- ironic.drivers.modules.ipxe
 - module, 2862
- ironic.drivers.modules.irmc
 - module, 2330
- ironic.drivers.modules.irmc.bios
 - module, 2260
- ironic.drivers.modules.irmc.boot
 - module, 2267
- ironic.drivers.modules.irmc.common
 - module, 2283
- ironic.drivers.modules.irmc.inspect
 - module, 2291
- ironic.drivers.modules.irmc.management
 - module, 2297
- ironic.drivers.modules.irmc.power
 - module, 2316
- ironic.drivers.modules.irmc.raid
 - module, 2325
- ironic.drivers.modules.network
 - module, 2379
- ironic.drivers.modules.network.common
 - module, 2330
- ironic.drivers.modules.network.flat
 - module, 2348
- ironic.drivers.modules.network.neutron
 - module, 2358
- ironic.drivers.modules.network.noop
 - module, 2369
- ironic.drivers.modules.noop
 - module, 2862
- ironic.drivers.modules.noop_mgmt
 - module, 2887
- ironic.drivers.modules.pxe
 - module, 2901
- ironic.drivers.modules.pxe_base
 - module, 2914
- ironic.drivers.modules.redfish
 - module, 2490
- ironic.drivers.modules.redfish.bios
 - module, 2379
- ironic.drivers.modules.redfish.boot
 - module, 2388
- ironic.drivers.modules.redfish.inspect
 - module, 2403
- ironic.drivers.modules.redfish.management
 - module, 2408
- ironic.drivers.modules.redfish.power
 - module, 2448
- ironic.drivers.modules.redfish.raid
 - module, 2457
- ironic.drivers.modules.redfish.utils
 - module, 2479
- ironic.drivers.modules.redfish.vendor
 - module, 2486
- ironic.drivers.modules.snmp
 - module, 2924
- ironic.drivers.modules.storage
 - module, 2507
- ironic.drivers.modules.storage.cinder
 - module, 2490
- ironic.drivers.modules.storage.external
 - module, 2496
- ironic.drivers.modules.storage.noop
 - module, 2502
- ironic.drivers.modules.xclarity
 - module, 2523
- ironic.drivers.modules.xclarity.common
 - module, 2507
- ironic.drivers.modules.xclarity.management
 - module, 2512
- ironic.drivers.modules.xclarity.power
 - module, 2518
- ironic.drivers.redfish
 - module, 3135
- ironic.drivers.snmp

- module, 3137
- ironic.drivers.utils
 - module, 3137
- ironic.drivers.xclarity
 - module, 3152
- ironic.objects
 - module, 3406
- ironic.objects.allocation
 - module, 3153
- ironic.objects.base
 - module, 3168
- ironic.objects.bios
 - module, 3182
- ironic.objects.chassis
 - module, 3197
- ironic.objects.conductor
 - module, 3208
- ironic.objects.deploy_template
 - module, 3217
- ironic.objects.deployment
 - module, 3233
- ironic.objects.fields
 - module, 3242
- ironic.objects.indirection
 - module, 3250
- ironic.objects.node
 - module, 3262
- ironic.objects.notification
 - module, 3311
- ironic.objects.port
 - module, 3318
- ironic.objects.portgroup
 - module, 3340
- ironic.objects.trait
 - module, 3357
- ironic.objects.volume_connector
 - module, 3367
- ironic.objects.volume_target
 - module, 3386
- ironic.version
 - module, 3406
- ironic-dbsync command line option
 - config-dir <DIR>, 537
 - config-file <PATH>, 537
 - debug, 537
 - help, 537
 - version, 537
 - d, 537
 - h, 537
 - create_schema,, 537
 - online_data_migrations, 537
 - revision, 537
 - stamp, 537
 - upgrade, 537
 - version, 537
- IronicBase (class in *ironic.db.sqlalchemy.models*), 1771
- IronicCORS (class in *ironic.api.app*), 1063
- IronicObject (class in *ironic.objects.base*), 3168
- IronicObjectIndirectionAPI (class in *ironic.objects.indirection*), 3250
- IronicObjectListBase (class in *ironic.objects.base*), 3178
- IronicObjectRegistry (class in *ironic.objects.base*), 3179
- IronicObjectSerializer (class in *ironic.objects.base*), 3179
- is_agent_token_pregenerated() (in module *ironic.conductor.utils*), 1586
- is_agent_token_present() (in module *ironic.conductor.utils*), 1587
- is_agent_token_valid() (in module *ironic.conductor.utils*), 1588
- is_anaconda_deploy() (in module *ironic.drivers.modules.deploy_utils*), 2673
- is_equivalent() (in module *ironic.conductor.steps*), 1548
- is_fast_track() (in module *ironic.conductor.utils*), 1589
- is_glance_image() (in module *ironic.common.glance_service.service_utils*), 1091
- is_hostname_safe() (in module *ironic.common.utils*), 1349
- is_image_active() (in module *ironic.common.glance_service.service_utils*), 1091
- is_image_available() (in module *ironic.common.glance_service.service_utils*), 1092
- is_ipxe_enabled() (in module *ironic.common.pxe_utils*), 1292
- is_iscsi_boot() (in module *ironic.drivers.modules.deploy_utils*), 2674
- is_memory_insufficient() (in module *ironic.common.utils*), 1351
- is_path_removed() (in module *ironic.api.controllers.v1.utils*), 1003
- is_path_updated() (in module *ironic.api.controllers.v1.utils*), 1004

- is_ramdisk_deploy() (in module *ironic.drivers.modules.deploy_utils*), 2674
- is_regex_string_in_file() (in module *ironic.common.utils*), 1353
- is_secure_boot_requested() (in module *ironic.drivers.modules.boot_mode_utils*), 2636
- is_smartnic (*ironic.db.sqlalchemy.models.Port* attribute), 1777
- is_smartnic (*ironic.objects.port.Port* property), 3326
- is_smartnic (*ironic.objects.port.PortCRUDPayload* property), 3339
- is_smartnic_port() (in module *ironic.common.neutron*), 1246
- is_software_raid() (in module *ironic.drivers.modules.deploy_utils*), 2674
- is_stable() (*ironic.common.fsm.FSM* method), 1181
- is_trusted_boot_requested() (in module *ironic.drivers.modules.boot_mode_utils*), 2638
- is_valid_datapath_id() (in module *ironic.common.utils*), 1353
- is_valid_logical_name() (in module *ironic.api.controllers.v1.utils*), 1005
- is_valid_logical_name() (in module *ironic.common.utils*), 1354
- is_valid_no_proxy() (in module *ironic.common.utils*), 1355
- is_valid_node_name() (in module *ironic.api.controllers.v1.utils*), 1005
- is_volume_attached() (in module *ironic.common.cinder*), 1126
- is_volume_available() (in module *ironic.common.cinder*), 1127
- is_whole_disk_image() (in module *ironic.common.images*), 1214
- ISCSIBOOT (in module *ironic.common.boot_devices*), 1112
- issue_startup_warnings() (in module *ironic.cmd.conductor*), 1081
- iswsmeffunction() (in module *ironic.api.functions*), 1068
- items() (*ironic.common.driver_factory.BaseDriverFactory* method), 1136
- iter_nodes() (*ironic.conductor.base_manager.BaseConductorManager* method), 1380
- J**
- JsonExtensionMiddleware (class in *ironic.api.middleware*), 1062
- JsonExtensionMiddleware (class in *ironic.api.middleware.json_ext*), 1060
- K**
- kernel_ref (*ironic.objects.deployment.Deployment* property), 3238
- KeystoneFailure, 1165
- KeystoneUnauthorized, 1166
- known_good_state() (*ironic.drivers.modules.drac.management.DracRedfishManager* method), 1993
- known_good_state() (*ironic.drivers.modules.drac.management.DracWSManManager* method), 2000
- ks_exceptions() (in module *ironic.common.keystone*), 1222
- L**
- last_error (*ironic.db.sqlalchemy.models.Allocation* attribute), 1766
- last_error (*ironic.db.sqlalchemy.models.Node* attribute), 1773
- last_error (*ironic.objects.allocation.Allocation* property), 3160
- last_error (*ironic.objects.allocation.AllocationCRUDPayload* property), 3167
- last_error (*ironic.objects.node.Node* property), 3272
- last_error (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- last_error (*ironic.objects.node.NodeCRUDPayload* property), 3286
- last_error (*ironic.objects.node.NodePayload* property), 3299
- last_error (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- last_error (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
- LEGACY_BIOS (in module *ironic.common.boot_modes*), 1114
- lessee (*ironic.db.sqlalchemy.models.Node* attribute), 1773
- lessee (*ironic.objects.node.Node* property), 3272
- lessee (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- lessee (*ironic.objects.node.NodeCRUDPayload* property), 3286
- lessee (*ironic.objects.node.NodePayload* property), 3299

[lessee \(*ironic.objects.node.NodeSetPowerStatePayload* class method\), 3393](#)
[property\), 3304](#) [list_by_names\(\)](#)
[lessee \(*ironic.objects.node.NodeSetProvisionStatePayload* \(*ironic.objects.deploy_template.DeployTemplate*
\[property\\), 3310\]\(#\) \[class method\\), 3226\]\(#\)
\[level \\(*ironic.objects.allocation.AllocationCRUDNotification* \\(*ironic.objects.port.Port*
\\[property\\\), 3166\\]\\(#\\) \\[class method\\\), 3328\\]\\(#\\)
\\[level \\\(*ironic.objects.chassis.ChassisCRUDNotification* \\\(*ironic.objects.portgroup.Portgroup*
\\\[property\\\\), 3207\\\]\\\(#\\\) \\\[list_by_node_id\\\\(\\\\)\\\]\\\(#\\\)
\\\[level \\\\(*ironic.objects.deploy_template.DeployTemplateCRUDNotification* \\\\(*ironic.objects.portgroup.Portgroup*
\\\\[property\\\\\), 3232\\\\]\\\\(#\\\\) \\\\[list_by_node_id\\\\\(\\\\\)\\\\]\\\\(#\\\\)
\\\\[level \\\\\(*ironic.objects.node.NodeConsoleNotification* \\\\\(*ironic.objects.volume_connector.VolumeConnector*
\\\\\[property\\\\\\), 3288\\\\\]\\\\\(#\\\\\) \\\\\[class method\\\\\\), 3377\\\\\]\\\\\(#\\\\\)
\\\\\[level \\\\\\(*ironic.objects.node.NodeCorrectedPowerStateNotification* \\\\\\(*ironic.objects.volume_target.VolumeTarget*
\\\\\\[property\\\\\\\), 3289\\\\\\]\\\\\\(#\\\\\\) \\\\\\[list_by_node_id\\\\\\\(\\\\\\\)\\\\\\]\\\\\\(#\\\\\\)
\\\\\\[level \\\\\\\(*ironic.objects.node.NodeCRUDNotification* class method\\\\\\\), 3395\\\\\\]\\\\\\(#\\\\\\)
\\\\\\[property\\\\\\\), 3281\\\\\\]\\\\\\(#\\\\\\) \\\\\\[list_by_portgroup_id\\\\\\\(\\\\\\\)\\\\\\]\\\\\\(#\\\\\\)
\\\\\\[level \\\\\\\(*ironic.objects.node.NodeMaintenanceNotification* \\\\\\\(*ironic.objects.port.Port* class method\\\\\\\),
\\\\\\\[property\\\\\\\\), 3295\\\\\\\]\\\\\\\(#\\\\\\\) \\\\\\\[3331\\\\\\\]\\\\\\\(#\\\\\\\)
\\\\\\\[level \\\\\\\\(*ironic.objects.node.NodeSetPowerStateNotification* \\\\\\\\(*ironic.objects.volume_target.VolumeTarget*
\\\\\\\\[property\\\\\\\\\), 3301\\\\\\\\]\\\\\\\\(#\\\\\\\\) \\\\\\\\[list_by_volume_id\\\\\\\\\(\\\\\\\\\)\\\\\\\\]\\\\\\\\(#\\\\\\\\)
\\\\\\\\[level \\\\\\\\\(*ironic.objects.node.NodeSetProvisionStateNotification* class method\\\\\\\\\), 3398\\\\\\\\]\\\\\\\\(#\\\\\\\\)
\\\\\\\\[property\\\\\\\\\), 3306\\\\\\\\]\\\\\\\\(#\\\\\\\\) \\\\\\\\[list_conductor_hardware_interfaces\\\\\\\\\(\\\\\\\\\)\\\\\\\\]\\\\\\\\(#\\\\\\\\)
\\\\\\\\[level \\\\\\\\\(*ironic.objects.notification.NotificationBase* \\\\\\\\\(*ironic.db.api.Connection* method\\\\\\\\\), 1872\\\\\\\\]\\\\\\\\(#\\\\\\\\)
\\\\\\\\[property\\\\\\\\\), 3315\\\\\\\\]\\\\\\\\(#\\\\\\\\) \\\\\\\\[list_conductor_hardware_interfaces\\\\\\\\\(\\\\\\\\\)\\\\\\\\]\\\\\\\\(#\\\\\\\\)
\\\\\\\\[level \\\\\\\\\(*ironic.objects.port.PortCRUDNotification* \\\\\\\\\(*ironic.db.sqlalchemy.api.Connection*
\\\\\\\\\[property\\\\\\\\\\), 3337\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\) \\\\\\\\\[method\\\\\\\\\\), 1719\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[level \\\\\\\\\\(*ironic.objects.portgroup.PortgroupCRUDNotification* \\\\\\\\\\(*ironic.api.controllers.v1.allocation*, 811\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[property\\\\\\\\\\), 3355\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\) \\\\\\\\\[list_convert_with_links\\\\\\\\\\(\\\\\\\\\\) \\\\\\\\\\(in module\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[level \\\\\\\\\\(*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* \\\\\\\\\\(*ironic.api.controllers.v1.chassis*\\\\\\\\\\), 819\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[property\\\\\\\\\\), 3384\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\) \\\\\\\\\[list_convert_with_links\\\\\\\\\\(\\\\\\\\\\) \\\\\\\\\\(in module\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[level \\\\\\\\\\(*ironic.objects.volume_target.VolumeTargetCRUDNotification* \\\\\\\\\\(*ironic.api.controllers.v1.collection*\\\\\\\\\\), 819\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[property\\\\\\\\\\), 3405\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\) \\\\\\\\\[list_convert_with_links\\\\\\\\\\(\\\\\\\\\\) \\\\\\\\\\(in module\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)
\\\\\\\\\[list\\\\\\\\\\(\\\\\\\\\\) \\\\\\\\\\(*ironic.objects.allocation.Allocation* class \\\\\\\\\\[list_convert_with_links\\\\\\\\\\\(\\\\\\\\\\\) \\\\\\\\\\\(in module\\\\\\\\\\]\\\\\\\\\\(#\\\\\\\\\\)
\\\\\\\\\\[method\\\\\\\\\\\), 3160\\\\\\\\\\]\\\\\\\\\\(#\\\\\\\\\\) \\\\\\\\\\[ironic.api.controllers.v1.conductor\\\\\\\\\\\), 826\\\\\\\\\\]\\\\\\\\\\(#\\\\\\\\\\)
\\\\\\\\\\[list\\\\\\\\\\\(\\\\\\\\\\\) \\\\\\\\\\\(*ironic.objects.chassis.Chassis* class \\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\(\\\\\\\\\\\\) \\\\\\\\\\\\(in module\\\\\\\\\\\]\\\\\\\\\\\(#\\\\\\\\\\\)
\\\\\\\\\\\[method\\\\\\\\\\\\), 3202\\\\\\\\\\\]\\\\\\\\\\\(#\\\\\\\\\\\) \\\\\\\\\\\[ironic.api.controllers.v1.deploy_template\\\\\\\\\\\\),\\\\\\\\\\\]\\\\\\\\\\\(#\\\\\\\\\\\)
\\\\\\\\\\\[list\\\\\\\\\\\\(\\\\\\\\\\\\) \\\\\\\\\\\\(*ironic.objects.conductor.Conductor* class \\\\\\\\\\\\[832\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[method\\\\\\\\\\\\\), 3210\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\) \\\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(in module\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[list\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(*ironic.objects.deploy_template.DeployTemplate* *ironic.api.controllers.v1.driver*\\\\\\\\\\\\\), 841\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[class method\\\\\\\\\\\\\), 3224\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\) \\\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(in module\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[list\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(*ironic.objects.deployment.Deployment* *ironic.api.controllers.v1.port*\\\\\\\\\\\\\), 915\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[class method\\\\\\\\\\\\\), 3238\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\) \\\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(in module\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)
\\\\\\\\\\\\[list\\\\\\\\\\\\\(\\\\\\\\\\\\\) \\\\\\\\\\\\\(*ironic.objects.node.Node* class method\\\\\\\\\\\\\), \\\\\\\\\\\\\[ironic.api.controllers.v1.portgroup\\\\\\\\\\\\\\), 924\\\\\\\\\\\\\]\\\\\\\\\\\\\(#\\\\\\\\\\\\\)
\\\\\\\\\\\\\[3272\\\\\\\\\\\\\]\\\\\\\\\\\\\(#\\\\\\\\\\\\\) \\\\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\\\(\\\\\\\\\\\\\\) \\\\\\\\\\\\\\(in module\\\\\\\\\\\\\]\\\\\\\\\\\\\(#\\\\\\\\\\\\\)
\\\\\\\\\\\\\[list\\\\\\\\\\\\\\(\\\\\\\\\\\\\\) \\\\\\\\\\\\\\(*ironic.objects.port.Port* class method\\\\\\\\\\\\\\), \\\\\\\\\\\\\\[ironic.api.controllers.v1.volume_connector\\\\\\\\\\\\\\\),\\\\\\\\\\\\\\]\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\[3326\\\\\\\\\\\\\\]\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\) \\\\\\\\\\\\\\[1038\\\\\\\\\\\\\\]\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\[list\\\\\\\\\\\\\\\(\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\(*ironic.objects.portgroup.Portgroup* class \\\\\\\\\\\\\\\[list_convert_with_links\\\\\\\\\\\\\\\\(\\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\\(in module\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\\[method\\\\\\\\\\\\\\\\), 3347\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\[ironic.api.controllers.v1.volume_target\\\\\\\\\\\\\\\\),\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\\[list\\\\\\\\\\\\\\\\(\\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\\(*ironic.objects.volume_connector.VolumeConnector* \\\\\\\\\\\\\\\\[1053\\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\\\[class method\\\\\\\\\\\\\\\\\), 3375\\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\\[list_hardware_type_interfaces\\\\\\\\\\\\\\\\\(\\\\\\\\\\\\\\\\\)\\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\\)
\\\\\\\\\\\\\\\\[list\\\\\\\\\\\\\\\\\(\\\\\\\\\\\\\\\\\) \\\\\\\\\\\\\\\\\(*ironic.objects.volume_target.VolumeTarget* \\\\\\\\\\\\\\\\\(*ironic.db.api.Connection* method\\\\\\\\\\\\\\\\\),\\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\\)\\\\\\\\\\\\\\\]\\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\\)\\\\\\\\\\\\\\]\\\\\\\\\\\\\\(#\\\\\\\\\\\\\\)\\\\\\\\\\\\\]\\\\\\\\\\\\\(#\\\\\\\\\\\\\)\\\\\\\\\\\\]\\\\\\\\\\\\(#\\\\\\\\\\\\)\\\\\\\\\\\]\\\\\\\\\\\(#\\\\\\\\\\\)\\\\\\\\\\]\\\\\\\\\\(#\\\\\\\\\\)\\\\\\\\\]\\\\\\\\\(#\\\\\\\\\)\\\\\\\\]\\\\\\\\(#\\\\\\\\)\\\\\\\]\\\\\\\(#\\\\\\\)\\\\\\]\\\\\\(#\\\\\\)\\\\\]\\\\\(#\\\\\)\\\\]\\\\(#\\\\)\\\]\\\(#\\\)\\]\\(#\\)\]\(#\)](#)

- 1873
 - `list_hardware_type_interfaces()` (*ironic.db.sqlalchemy.api.Connection method*), 1719
 - `list_opts()` (*in module ironic.conf.cinder*), 1622
 - `list_opts()` (*in module ironic.conf.default*), 1623
 - `list_opts()` (*in module ironic.conf.glance*), 1623
 - `list_opts()` (*in module ironic.conf.inspector*), 1624
 - `list_opts()` (*in module ironic.conf.neutron*), 1625
 - `list_opts()` (*in module ironic.conf.nova*), 1625
 - `list_opts()` (*in module ironic.conf.opts*), 1625
 - `list_opts()` (*in module ironic.conf.service_catalog*), 1627
 - `list_opts()` (*in module ironic.conf.swift*), 1627
 - `list_physical_disks()` (*in module ironic.drivers.modules.drac.raid*), 2041
 - `list_policies()` (*in module ironic.common.policy*), 1266
 - `list_raid_controllers()` (*in module ironic.drivers.modules.drac.raid*), 2042
 - `list_raid_settings()` (*in module ironic.drivers.modules.drac.raid*), 2043
 - `list_unfinished_jobs()` (*in module ironic.drivers.modules.drac.job*), 1983
 - `list_unfinished_jobs()` (*in module ironic.drivers.modules.drac.vendor_passthru.WBEMVendorPassthru method*), 2058
 - `list_virtual_disks()` (*in module ironic.drivers.modules.drac.raid*), 2044
 - `ListOfFlexibleDictsField` (*class in ironic.objects.fields*), 3244
 - `ListOfObjectsField` (*class in ironic.objects.fields*), 3244
 - `ListOfStringsField` (*class in ironic.objects.fields*), 3244
 - `load_driver()` (*in module ironic.conductor.task_manager.TaskManager method*), 1563
 - `local_link_connection` (*in module ironic.db.sqlalchemy.models.Port attribute*), 1777
 - `local_link_connection` (*in module ironic.objects.port.Port property*), 3333
 - `local_link_connection` (*in module ironic.objects.port.PortCRUDPayload property*), 3339
 - `local_link_normalize()` (*in module ironic.api.controllers.v1.utils*), 1006
 - `log_and_raise_deployment_error()` (*in module ironic.drivers.modules.agent_base*), 2584
 - `logical_disk_properties()` (*in module ironic.api.controllers.v1.driver.DriverRaidController method*), 835
 - `LOOKUP_ALLOWED_STATES` (*in module ironic.common.states*), 1326
 - `lookup_allowed_states` (*in module ironic.api.controllers.v1.ramdisk.LookupController property*), 930
 - `LookupController` (*class in ironic.api.controllers.v1.ramdisk*), 928
 - `lower_bound` (*in module ironic.db.sqlalchemy.models.BIOSSetting attribute*), 1767
 - `lower_bound` (*in module ironic.objects.bios.BIOSSetting property*), 3187
- ## M
- `mac_address()` (*in module ironic.common.args*), 1099
 - `MACAddress` (*class in ironic.objects.fields*), 3244
 - `MACAddressField` (*class in ironic.objects.fields*), 3246
 - `MACAlreadyExists`, 1166
 - `main()` (*in module ironic.cmd.api*), 1081
 - `main()` (*in module ironic.cmd.wbem_vendor_passthru*), 1081
 - `main()` (*in module ironic.cmd.dbsync*), 1083
 - `main()` (*in module ironic.cmd.status*), 1084
 - `maintenance` (*in module ironic.api.controllers.v1.node.NodesController attribute*), 893
 - `maintenance` (*in module ironic.db.sqlalchemy.models.Node attribute*), 1773
 - `maintenance` (*in module ironic.objects.node.Node property*), 3274
 - `maintenance` (*in module ironic.objects.node.NodeCorrectedPowerStatePayload property*), 3293
 - `maintenance` (*in module ironic.objects.node.NodeCRUDPayload property*), 3286
 - `maintenance` (*in module ironic.objects.node.NodePayload property*), 3299
 - `maintenance` (*in module ironic.objects.node.NodeSetPowerStatePayload property*), 3304
 - `maintenance` (*in module ironic.objects.node.NodeSetProvisionStatePayload property*), 3310
 - `maintenance_reason` (*in module ironic.db.sqlalchemy.models.Node*

- attribute), 1773
- maintenance_reason (*ironic.objects.node.Node* property), 3274
- maintenance_reason (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- maintenance_reason (*ironic.objects.node.NodeCRUDPayload* property), 3286
- maintenance_reason (*ironic.objects.node.NodePayload* property), 3299
- maintenance_reason (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- maintenance_reason (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
- make_link() (in module *ironic.api.controllers.link*), 1056
- make_persistent_password_file() (in module *ironic.drivers.modules.console_utils*), 2644
- make_salt() (in module *ironic.conductor.utils*), 1591
- MANAGEABLE (in module *ironic.common.states*), 1326
- management (*ironic.api.controllers.v1.node.NodesController* attribute), 893
- management (*ironic.drivers.base.BareDriver* attribute), 2961
- management_interface (*ironic.db.sqlalchemy.models.Node* attribute), 1773
- management_interface (*ironic.objects.node.Node* property), 3274
- management_interface (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- management_interface (*ironic.objects.node.NodeCRUDPayload* property), 3286
- management_interface (*ironic.objects.node.NodePayload* property), 3299
- management_interface (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- management_interface (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
- ManagementInterface (class in *ironic.drivers.base*), 3003
- mandatory (*ironic.api.functions.FunctionArgument* attribute), 1065
- ManualManagementHardware (class in *ironic.drivers.generic*), 3118
- mask_secrets() (in module *ironic.objects.notification*), 3317
- max_length (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
- max_length (*ironic.objects.bios.BIOSSetting* property), 3187
- max_string (*ironic.api.controllers.base.Version* attribute), 1054
- max_version() (in module *ironic.objects.base*), 3181
- max_version_string() (in module *ironic.api.controllers.v1.versions*), 1022
- memoize() (in module *ironic.drivers.modules.snmp*), 2950
- metadata (*ironic.db.sqlalchemy.models.IronicBase* attribute), 1772
- metadata (*ironic.drivers.base.VendorMetadata* attribute), 3095
- method (*ironic.drivers.base.VendorMetadata* attribute), 3095
- methods() (*ironic.api.controllers.v1.driver.DriverPassthruController* method), 834
- methods() (*ironic.api.controllers.v1.node.NodeVendorPassthruController* method), 876
- METRICS (in module *ironic.drivers.modules.irmc.inspect*), 2295
- METRICS (in module *ironic.drivers.modules.irmc.power*), 2325
- min_length (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
- min_length (*ironic.objects.bios.BIOSSetting* property), 3188
- min_string (*ironic.api.controllers.base.Version* attribute), 1054
- min_version_string() (in module *ironic.api.controllers.v1.versions*), 1022
- MissingParameterValue, 1166
- MixinVendorInterface (class in *ironic.drivers.utils*), 3137
- Node (*ironic.db.sqlalchemy.models.Portgroup* attribute), 1773

- tribute), 1778
- mode (*ironic.objects.portgroup.Portgroup* property), 3352
- mode (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 3357
- model_query() (in module *ironic.db.sqlalchemy.api*), 1761
- module
 - ironic, 3407
 - ironic.api, 1081
 - ironic.api.app, 1063
 - ironic.api.config, 1064
 - ironic.api.controllers, 1059
 - ironic.api.controllers.base, 1054
 - ironic.api.controllers.link, 1056
 - ironic.api.controllers.root, 1056
 - ironic.api.controllers.v1, 1053
 - ironic.api.controllers.v1.allocation, 805
 - ironic.api.controllers.v1.bios, 811
 - ironic.api.controllers.v1.chassis, 813
 - ironic.api.controllers.v1.collection, 819
 - ironic.api.controllers.v1.conductor, 822
 - ironic.api.controllers.v1.deploy_template, 826
 - ironic.api.controllers.v1.driver, 833
 - ironic.api.controllers.v1.event, 842
 - ironic.api.controllers.v1.node, 843
 - ironic.api.controllers.v1.notification_utils, 901
 - ironic.api.controllers.v1.port, 905
 - ironic.api.controllers.v1.portgroup, 916
 - ironic.api.controllers.v1.ramdisk, 924
 - ironic.api.controllers.v1.utils, 931
 - ironic.api.controllers.v1.versions, 1022
 - ironic.api.controllers.v1.volume, 1023
 - ironic.api.controllers.v1.volume_connectors, 1023
 - ironic.api.controllers.v1.volume_target, 1038
 - ironic.api.controllers.version, 1057
 - ironic.api.functions, 1064
 - ironic.api.hooks, 1071
 - ironic.api.method, 1079
 - ironic.api.middleware, 1061
 - ironic.api.middleware.auth_public_routes, 1059
 - ironic.api.middleware.json_ext, 1060
 - ironic.api.middleware.parsable_error, 1060
 - ironic.api.wsgi, 1080
 - ironic.cmd, 1084
 - ironic.cmd.api, 1081
 - ironic.cmd.conductor, 1081
 - ironic.cmd.dbsync, 1081
 - ironic.cmd.status, 1083
 - ironic.common, 1372
 - ironic.common.args, 1093
 - ironic.common.boot_devices, 1110
 - ironic.common.boot_modes, 1113
 - ironic.common.cinder, 1114
 - ironic.common.components, 1128
 - ironic.common.config, 1130
 - ironic.common.context, 1130
 - ironic.common.dhcp_factory, 1132
 - ironic.common.driver_factory, 1136
 - ironic.common.exception, 1152
 - ironic.common.faults, 1176
 - ironic.common.fsm, 1177
 - ironic.common.glance_service, 1093
 - ironic.common.glance_service.image_service, 1084
 - ironic.common.glance_service.service_utils, 1091
 - ironic.common.hash_ring, 1183
 - ironic.common.i18n, 1183
 - ironic.common.image_service, 1184
 - ironic.common.images, 1197
 - ironic.common.indicator_states, 1216
 - ironic.common.keystone, 1216
 - ironic.common.molds, 1223
 - ironic.common.network, 1226
 - ironic.common.neutron, 1233
 - ironic.common.nova, 1260
 - ironic.common.policy, 1262
 - ironic.common.profiler, 1266
 - ironic.common.pxe_utils, 1269
 - ironic.common.raid, 1300
 - ironic.common.release_mappings, 1307
 - ironic.common.rpc, 1309
 - ironic.common.rpc_service, 1313
 - ironic.common.service, 1315
 - ironic.common.states, 1315
 - ironic.common.swift, 1334

ironic.common.utils, 1342
ironic.common.wsgi_service, 1370
ironic.conductor, 1620
ironic.conductor.allocations, 1372
ironic.conductor.base_manager, 1377
ironic.conductor.cleaning, 1383
ironic.conductor.deployments, 1388
ironic.conductor.manager, 1393
ironic.conductor.notification_utils,
1412
ironic.conductor.rpcapi, 1418
ironic.conductor.steps, 1548
ironic.conductor.task_manager, 1554
ironic.conductor.utils, 1573
ironic.conf, 1628
ironic.conf.agent, 1620
ironic.conf.anaconda, 1621
ironic.conf.ansible, 1621
ironic.conf.api, 1621
ironic.conf.audit, 1621
ironic.conf.auth, 1621
ironic.conf.cinder, 1622
ironic.conf.conductor, 1623
ironic.conf.console, 1623
ironic.conf.database, 1623
ironic.conf.default, 1623
ironic.conf.deploy, 1623
ironic.conf.dhcp, 1623
ironic.conf.drac, 1623
ironic.conf.glance, 1623
ironic.conf.healthcheck, 1624
ironic.conf.ibm, 1624
ironic.conf.ilo, 1624
ironic.conf.inspector, 1624
ironic.conf.ipmi, 1624
ironic.conf.irmc, 1624
ironic.conf.metrics, 1624
ironic.conf.metrics_statsd, 1624
ironic.conf.molds, 1625
ironic.conf.neutron, 1625
ironic.conf.nova, 1625
ironic.conf.opts, 1625
ironic.conf.pxe, 1627
ironic.conf.redfish, 1627
ironic.conf.service_catalog, 1627
ironic.conf.snmp, 1627
ironic.conf.swift, 1627
ironic.conf.xclarity, 1628
ironic.db, 1911
ironic.db.api, 1781
ironic.db.migration, 1911
ironic.db.sqlalchemy, 1781
ironic.db.sqlalchemy.api, 1628
ironic.db.sqlalchemy.migration, 1762
ironic.db.sqlalchemy.models, 1765
ironic.dhcp, 1933
ironic.dhcp.base, 1911
ironic.dhcp.neutron, 1919
ironic.dhcp.none, 1926
ironic.drivers, 3153
ironic.drivers.base, 2950
ironic.drivers.drac, 3110
ironic.drivers.fake_hardware, 3112
ironic.drivers.generic, 3116
ironic.drivers.hardware_type, 3120
ironic.drivers.ibm, 3125
ironic.drivers.ilo, 3126
ironic.drivers.intel_ipmi, 3130
ironic.drivers.ipmi, 3131
ironic.drivers.irmc, 3132
ironic.drivers.modules, 2950
ironic.drivers.modules.agent, 2523
ironic.drivers.modules.agent_base,
2553
ironic.drivers.modules.agent_client,
2590
ironic.drivers.modules.agent_power,
2625
ironic.drivers.modules.ansible, 1943
ironic.drivers.modules.ansible.deploy,
1933
ironic.drivers.modules.boot_mode_utils,
2634
ironic.drivers.modules.console_utils,
2643
ironic.drivers.modules.deploy_utils,
2650
ironic.drivers.modules.drac, 2063
ironic.drivers.modules.drac.bios,
1943
ironic.drivers.modules.drac.boot,
1967
ironic.drivers.modules.drac.common,
1972
ironic.drivers.modules.drac.inspect,
1975
ironic.drivers.modules.drac.job,
1981
ironic.drivers.modules.drac.management,
1985
ironic.drivers.modules.drac.power,
2007

`ironic.drivers.modules.drac.raid`, 2016

`ironic.drivers.modules.drac.utils`, 2048

`ironic.drivers.modules.drac.vendor_passthru`, 2051

`ironic.drivers.modules.fake`, 2695

`ironic.drivers.modules.ibmc`, 2106

`ironic.drivers.modules.ibmc.management`, 2063

`ironic.drivers.modules.ibmc.mappings`, 2079

`ironic.drivers.modules.ibmc.power`, 2079

`ironic.drivers.modules.ibmc.raid`, 2089

`ironic.drivers.modules.ibmc.utils`, 2097

`ironic.drivers.modules.ibmc.vendor`, 2099

`ironic.drivers.modules.ilo`, 2259

`ironic.drivers.modules.ilo.bios`, 2106

`ironic.drivers.modules.ilo.boot`, 2112

`ironic.drivers.modules.ilo.common`, 2149

`ironic.drivers.modules.ilo.console`, 2192

`ironic.drivers.modules.ilo.firmware_processor`, 2194

`ironic.drivers.modules.ilo.inspect`, 2200

`ironic.drivers.modules.ilo.management`, 2204

`ironic.drivers.modules.ilo.power`, 2239

`ironic.drivers.modules.ilo.raid`, 2247

`ironic.drivers.modules.ilo.vendor`, 2254

`ironic.drivers.modules.image_cache`, 2787

`ironic.drivers.modules.image_utils`, 2793

`ironic.drivers.modules.inspect_utils`, 2811

`ironic.drivers.modules.inspector`, 2813

`ironic.drivers.modules.intel_ipmi`, 2260

`ironic.drivers.modules.intel_ipmi.management`, 2260

`ironic.drivers.modules.ipmitool`, 2816

`ironic.drivers.modules.ipxe`, 2862

`ironic.drivers.modules.irmc`, 2330

`ironic.drivers.modules.irmc.bios`, 2260

`ironic.drivers.modules.irmc.boot`, 2267

`ironic.drivers.modules.irmc.common`, 2283

`ironic.drivers.modules.irmc.inspect`, 2291

`ironic.drivers.modules.irmc.management`, 2297

`ironic.drivers.modules.irmc.power`, 2316

`ironic.drivers.modules.irmc.raid`, 2325

`ironic.drivers.modules.network`, 2379

`ironic.drivers.modules.network.common`, 2330

`ironic.drivers.modules.network.flat`, 2348

`ironic.drivers.modules.network.neutron`, 2358

`ironic.drivers.modules.network.noop`, 2369

`ironic.drivers.modules.noop`, 2862

`ironic.drivers.modules.noop_mgmt`, 2887

`ironic.drivers.modules.pxe`, 2901

`ironic.drivers.modules.pxe_base`, 2914

`ironic.drivers.modules.redfish`, 2490

`ironic.drivers.modules.redfish.bios`, 2379

`ironic.drivers.modules.redfish.boot`, 2388

`ironic.drivers.modules.redfish.inspect`, 2403

`ironic.drivers.modules.redfish.management`, 2408

`ironic.drivers.modules.redfish.power`, 2448

`ironic.drivers.modules.redfish.raid`, 2457

`ironic.drivers.modules.redfish.utils`, 2479

`ironic.drivers.modules.redfish.vendor`,

- 2486
 ironic.drivers.modules.snmp, 2924
 ironic.drivers.modules.storage, 2507
 ironic.drivers.modules.storage.cinder, 2490
 ironic.drivers.modules.storage.external, 2496
 ironic.drivers.modules.storage.noop, 2502
 ironic.drivers.modules.xclarity, 2523
 ironic.drivers.modules.xclarity.common, 2507
 ironic.drivers.modules.xclarity.management, 2512
 ironic.drivers.modules.xclarity.power, 2518
 ironic.drivers.redfish, 3135
 ironic.drivers.snmp, 3137
 ironic.drivers.utils, 3137
 ironic.drivers.xclarity, 3152
 ironic.objects, 3406
 ironic.objects.allocation, 3153
 ironic.objects.base, 3168
 ironic.objects.bios, 3182
 ironic.objects.chassis, 3197
 ironic.objects.conductor, 3208
 ironic.objects.deploy_template, 3217
 ironic.objects.deployment, 3233
 ironic.objects.fields, 3242
 ironic.objects.indirection, 3250
 ironic.objects.node, 3262
 ironic.objects.notification, 3311
 ironic.objects.port, 3318
 ironic.objects.portgroup, 3340
 ironic.objects.trait, 3357
 ironic.objects.volume_connector, 3367
 ironic.objects.volume_target, 3386
 ironic.version, 3406
 mount() (in module ironic.common.utils), 1357
- ## N
- name (ironic.api.functions.FunctionArgument attribute), 1065
 name (ironic.api.functions.FunctionDefinition attribute), 1067
 name (ironic.db.sqlalchemy.models.Allocation attribute), 1766
 name (ironic.db.sqlalchemy.models.BIOSSetting attribute), 1767
 name (ironic.db.sqlalchemy.models.DeployTemplate attribute), 1770
 name (ironic.db.sqlalchemy.models.Node attribute), 1774
 name (ironic.db.sqlalchemy.models.Port attribute), 1777
 name (ironic.db.sqlalchemy.models.Portgroup attribute), 1778
 name (ironic.objects.allocation.Allocation property), 3162
 name (ironic.objects.allocation.AllocationCRUDPayload property), 3167
 name (ironic.objects.bios.BIOSSetting property), 3188
 name (ironic.objects.deploy_template.DeployTemplate property), 3228
 name (ironic.objects.deploy_template.DeployTemplateCRUDPayload property), 3233
 name (ironic.objects.node.Node property), 3274
 name (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293
 name (ironic.objects.node.NodeCRUDPayload property), 3286
 name (ironic.objects.node.NodePayload property), 3299
 name (ironic.objects.node.NodeSetPowerStatePayload property), 3304
 name (ironic.objects.node.NodeSetProvisionStatePayload property), 3310
 name (ironic.objects.port.Port property), 3333
 name (ironic.objects.port.PortCRUDPayload property), 3339
 name (ironic.objects.portgroup.Portgroup property), 3352
 name (ironic.objects.portgroup.PortgroupCRUDPayload property), 3357
 name() (in module ironic.common.args), 1100
 names (ironic.common.driver_factory.BaseDriverFactory property), 1136
 need_power_on() (ironic.drivers.base.NetworkInterface method), 3051
 need_power_on() (ironic.drivers.modules.network.neutron.NeutronNetwork method), 2362
 network (ironic.drivers.base.BareDriver attribute), 2961
 network_data (ironic.db.sqlalchemy.models.Node attribute), 1774
 network_data (ironic.objects.node.Node property), 3274

- `network_data_schema()` (in module `ironic.api.controllers.v1.node`), 898
- `network_interface` (`ironic.db.sqlalchemy.models.Node` attribute), 1774
- `network_interface` (`ironic.objects.node.Node` property), 3274
- `network_interface` (`ironic.objects.node.NodeCorrectedPowerStatePayload` property), 3293
- `network_interface` (`ironic.objects.node.NodeCRUDPayload` property), 3286
- `network_interface` (`ironic.objects.node.NodePayload` property), 3299
- `network_interface` (`ironic.objects.node.NodeSetPowerStatePayload` property), 3304
- `network_interface` (`ironic.objects.node.NodeSetProvisionStatePayload` property), 3310
- `NetworkError`, 1166
- `NetworkInterface` (class in `ironic.drivers.base`), 3042
- `NetworkInterfaceFactory` (class in `ironic.common.driver_factory`), 1137
- `NeutronDHCPApi` (class in `ironic.dhcp.neutron`), 1919
- `NeutronNetwork` (class in `ironic.drivers.modules.network.neutron`), 2358
- `NeutronNetworkInterfaceMixin` (class in `ironic.common.neutron`), 1233
- `NeutronVIFPortIDMixin` (class in `ironic.drivers.modules.network.common`), 2330
- `NIC` (in module `ironic.common.components`), 1129
- `NoBIOS` (class in `ironic.drivers.modules.noop`), 2863
- `NoConsole` (class in `ironic.drivers.modules.noop`), 2870
- `NoConsolePid`, 1166
- `Node` (class in `ironic.db.sqlalchemy.models`), 1772
- `Node` (class in `ironic.objects.node`), 3262
- `node` (`ironic.conductor.task_manager.TaskManager` property), 1563
- `node` (`ironic.db.sqlalchemy.models.NodeTag` attribute), 1775
- `node` (`ironic.db.sqlalchemy.models.NodeTrait` attribute), 1776
- `node_cache_bios_settings()` (in module `ironic.conductor.utils`), 1591
- `node_cache_vendor()` (in module `ironic.conductor.utils`), 1592
- `NODE_CLASS_OID` (in module `ironic.drivers.modules.irmc.inspect`), 2296
- `node_convert_with_links()` (in module `ironic.api.controllers.v1.node`), 898
- `node_get_boot_mode()` (in module `ironic.conductor.utils`), 1592
- `node_id` (`ironic.db.sqlalchemy.models.Allocation` attribute), 1766
- `node_id` (`ironic.db.sqlalchemy.models.BIOSSetting` attribute), 1767
- `node_id` (`ironic.db.sqlalchemy.models.NodeTag` attribute), 1775
- `node_id` (`ironic.db.sqlalchemy.models.NodeTrait` attribute), 1776
- `node_id` (`ironic.db.sqlalchemy.models.Port` attribute), 1777
- `node_id` (`ironic.db.sqlalchemy.models.Portgroup` attribute), 1778
- `node_id` (`ironic.db.sqlalchemy.models.VolumeConnector` attribute), 1779
- `node_id` (`ironic.db.sqlalchemy.models.VolumeTarget` attribute), 1780
- `node_id` (`ironic.objects.allocation.Allocation` property), 3162
- `node_id` (`ironic.objects.bios.BIOSSetting` property), 3188
- `node_id` (`ironic.objects.port.Port` property), 3333
- `node_id` (`ironic.objects.portgroup.Portgroup` property), 3352
- `node_id` (`ironic.objects.trait.Trait` property), 3362
- `node_id` (`ironic.objects.volume_connector.VolumeConnector` property), 3380
- `node_id` (`ironic.objects.volume_target.VolumeTarget` property), 3400
- `node_list_convert_with_links()` (in module `ironic.api.controllers.v1.node`), 898
- `node_mapping` (`ironic.objects.deployment.Deployment` attribute), 3240
- `node_patch_schema()` (in module `ironic.api.controllers.v1.node`), 898
- `node_patch_validator()` (in module `ironic.api.controllers.v1.node`), 898
- `node_power_action()` (in module `ironic.conductor.utils`), 1594
- `node_sanitize()` (in module `ironic.api.controllers.v1.node`), 898

- node_schema() (in module *ironic.objects.node*), 3290
ironic.api.controllers.v1.node), 899
- node_set_boot_device() (in module *ironic.conductor.utils*), 1596
- node_set_boot_mode() (in module *ironic.conductor.utils*), 1598
- node_states_convert() (in module *ironic.api.controllers.v1.node*), 899
- node_tag_exists() (*ironic.db.api.Connection* method), 1873
- node_tag_exists() (*ironic.db.sqlalchemy.api.Connection* method), 1720
- node_trait_exists() (*ironic.db.api.Connection* method), 1874
- node_trait_exists() (*ironic.db.sqlalchemy.api.Connection* method), 1721
- node_uuid (*ironic.objects.allocation.AllocationCRUDPayload* property), 3167
- node_uuid (*ironic.objects.deployment.Deployment* property), 3240
- node_uuid (*ironic.objects.port.PortCRUDPayload* property), 3339
- node_uuid (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 3357
- node_uuid (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 3385
- node_uuid (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406
- node_validator() (in module *ironic.api.controllers.v1.node*), 899
- node_wait_for_power_state() (in module *ironic.conductor.utils*), 1601
- NodeAllocationController (class in *ironic.api.controllers.v1.allocation*), 810
- NodeAlreadyExists, 1167
- NodeAssociated, 1167
- NodeBiosController (class in *ironic.api.controllers.v1.bios*), 811
- NodeCleaningFailure, 1167
- NodeConsoleController (class in *ironic.api.controllers.v1.node*), 853
- NodeConsoleNotEnabled, 1168
- NodeConsoleNotification (class in *ironic.objects.node*), 3287
- NodeCorrectedPowerStateNotification (class in *ironic.objects.node*), 3288
- NodeCorrectedPowerStatePayload (class in *ironic.objects.node*), 3281
- NodeCRUDNotification (class in *ironic.objects.node*), 3281
- NodeCRUDPayload (class in *ironic.objects.node*), 3282
- NodeInMaintenance, 1168
- NodeIsRetired, 1168
- NodeLocked, 1168
- NodeMaintenanceController (class in *ironic.api.controllers.v1.node*), 854
- NodeMaintenanceFailure, 1168
- NodeMaintenanceNotification (class in *ironic.objects.node*), 3294
- NodeManagementController (class in *ironic.api.controllers.v1.node*), 856
- NodeNotFound, 1168
- NodeNotLocked, 1168
- NodePayload (class in *ironic.objects.node*), 3295
- NodeProtected, 1169
- NodeSetPowerStateNotification (class in *ironic.api.controllers.v1.chassis.ChassisController* attribute), 817
- NodesController (class in *ironic.api.controllers.v1.node*), 877
- NodeSetPowerStateNotification (class in *ironic.objects.node*), 3300
- NodeSetPowerStatePayload (class in *ironic.objects.node*), 3301
- NodeSetPowerStateNotification (class in *ironic.objects.node*), 3305
- NodeSetProvisionStatePayload (class in *ironic.objects.node*), 3306
- NodeStatesController (class in *ironic.api.controllers.v1.node*), 857
- NodeTag (class in *ironic.db.sqlalchemy.models*), 1775
- NodeTagNotFound, 1169
- NodeTrait (class in *ironic.db.sqlalchemy.models*), 1775
- NodeTraitNotFound, 1169
- NodeTraitsController (class in *ironic.api.controllers.v1.node*), 872
- NodeVendorPassthruController (class in *ironic.api.controllers.v1.node*), 876
- NodeVIFController (class in *ironic.api.controllers.v1.node*), 874
- NoDriversLoaded, 1166
- NoExceptionTracebackHook (class in *ironic.api.hooks*), 1075
- NoFreeConductorWorker, 1166
- NoFreeIPMITerminalPorts, 1167
- NoFreePhysicalPorts, 1167

- NoInspect (*class in ironic.drivers.modules.noop*), 2873
- non_vendor_interfaces (*ironic.drivers.base.BareDriver property*), 2962
- NoneDHCPApi (*class in ironic.dhcp.none*), 1926
- NoopManagement (*class in ironic.drivers.modules.noop_mgmt*), 2887
- NoopNetwork (*class in ironic.drivers.modules.network.noop*), 2369
- NoopStorage (*class in ironic.drivers.modules.storage.noop*), 2502
- NoRAID (*class in ironic.drivers.modules.noop*), 2875
- NoRescue (*class in ironic.drivers.modules.noop*), 2881
- normalize_mac() (*in module ironic.drivers.utils*), 3149
- NOSTATE (*in module ironic.common.states*), 1327
- NotAcceptable, 1169
- NotAuthorized, 1169
- NotFound, 1169
- NotificationBase (*class in ironic.objects.notification*), 3313
- NotificationLevel (*class in ironic.objects.fields*), 3246
- NotificationLevelField (*class in ironic.objects.fields*), 3247
- NotificationPayloadBase (*class in ironic.objects.notification*), 3315
- NotificationPayloadError, 1170
- NotificationPublisher (*class in ironic.objects.notification*), 3317
- NotificationSchemaKeyError, 1170
- NotificationSchemaObjectError, 1170
- NotificationStatus (*class in ironic.objects.fields*), 3247
- NotificationStatusField (*class in ironic.objects.fields*), 3247
- notify_conductor_resume_clean() (*in module ironic.conductor.utils*), 1602
- notify_conductor_resume_deploy() (*in module ironic.conductor.utils*), 1602
- notify_conductor_resume_operation() (*in module ironic.conductor.utils*), 1602
- NoValidDefaultForInterface, 1167
- NoValidHost, 1167
- NoVendor (*class in ironic.drivers.modules.noop*), 2884
- ## O
- obj (*ironic.api.controllers.v1.utils.PassthruResponse attribute*), 931
- OBJ_BASE_CLASS (*ironic.objects.base.IronicObjectSerializer attribute*), 3179
- OBJ_PROJECT_NAMESPACE (*ironic.objects.base.IronicObject attribute*), 3169
- obj_refresh() (*ironic.objects.base.IronicObject method*), 3176
- OBJ_SERIAL_NAMESPACE (*ironic.objects.base.IronicObject attribute*), 3169
- object (*ironic.objects.notification.EventType property*), 3312
- object_action() (*ironic.conductor.manager.ConductorManager method*), 1401
- object_action() (*ironic.conductor.rpcapi.ConductorAPI method*), 1503
- object_action() (*ironic.objects.indirection.IronicObjectIndirectionAPI method*), 3250
- object_backport_versions() (*ironic.conductor.manager.ConductorManager method*), 1403
- object_backport_versions() (*ironic.conductor.rpcapi.ConductorAPI method*), 1505
- object_backport_versions() (*ironic.objects.indirection.IronicObjectIndirectionAPI method*), 3253
- object_class_action() (*ironic.objects.indirection.IronicObjectIndirectionAPI method*), 3255
- object_class_action_versions() (*ironic.conductor.manager.ConductorManager method*), 1405
- object_class_action_versions() (*ironic.conductor.rpcapi.ConductorAPI method*), 1508
- object_class_action_versions() (*ironic.objects.indirection.IronicObjectIndirectionAPI method*), 3258
- object_to_dict() (*in module ironic.api.controllers.v1.utils*), 1006
- ObjectField (*class in ironic.objects.fields*), 3247

- objects (*ironic.objects.bios.BIOSSettingList* property), 3194
- objects (*ironic.objects.trait.TraitList* property), 3367
- OFF (in module *ironic.common.indicator_states*), 1216
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAten* attribute), 2929
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCRack* attribute), 2930
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU* attribute), 2931
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAten* attribute), 2932
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverBareMetal* attribute), 2936
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverCyberPower* attribute), 2937
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 2938
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverSimple* property), 2940
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 2941
- oid_enterprise (*ironic.drivers.modules.snmp.SNMPDriverBareMetal* attribute), 2933
- oid_poweroff (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 2938
- oid_poweron (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 2938
- oid_status (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 2939
- ON (in module *ironic.common.indicator_states*), 1216
- on_enter() (in module *ironic.common.states*), 1334
- on_exit() (in module *ironic.common.states*), 1334
- one_button_secure_erase() (*ironic.drivers.modules.ilo.management.IloManagement* method), 2206
- online (*ironic.db.sqlalchemy.models.Conductor* attribute), 1769
- online_data_migrations *ironic-dbsync* command line option, 537
- online_data_migrations command line option --help, 538
- online_data_migrations command line option --max-count <NUMBER>, 538
- option <MIGRATION.KEY=VALUE>, 538
- h, 538
- online_data_migrations() (*ironic.cmd.dbsync.DBCommand* method), 1083
- OperationNotPermitted, 1170
- opt_attrs (*ironic.drivers.base.BareDriver* property), 562
- or_valid() (in module *ironic.common.args*), 1083
- override_api_url() (in module *ironic.drivers.modules.image_utils*), 2798
- owner (*ironic.db.sqlalchemy.models.Allocation* attribute), 1766
- owner (*ironic.db.sqlalchemy.models.Node* attribute), 1774
- owner (*ironic.objects.allocation.Allocation* property), 3162
- owner (*ironic.objects.allocation.AllocationCRUDPayload* property), 3168
- owner (*ironic.objects.node.Node* property), 3274
- owner (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- owner (*ironic.objects.node.NodeCRUDPayload* property), 3286
- owner (*ironic.objects.node.NodePayload* property), 3299
- owner (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- owner (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310

P

- ParsableErrorMiddleware (class in *ironic.api.middleware*), 1063
- ParsableErrorMiddleware (class in *ironic.api.middleware.parsable_error*), 1061
- ping_pong() (in module *ironic.common.config*), 1130
- parse_driver_info() (in module *ironic.common.pxe_utils*), 1293
- parse_driver_info() (in module *ironic.drivers.modules.drac.common*), 1973
- parse_driver_info() (in module *ironic.drivers.modules.ibmutils*), 2098
- parse_driver_info() (in module

- `ironic.drivers.modules.ilo.boot`), 2145
- `parse_driver_info()` (in module `ironic.drivers.modules.ilo.common`), 2170
- `parse_driver_info()` (in module `ironic.drivers.modules.irmc.common`), 2288
- `parse_driver_info()` (in module `ironic.drivers.modules.redfish.utils`), 2484
- `parse_driver_info()` (in module `ironic.drivers.modules.xclarity.common`), 2509
- `parse_headers()` (`ironic.api.controllers.base.Version` static method), 1054
- `parse_image_id()` (in module `ironic.common.glance_service.service_utils`), 1092
- `parse_instance_info()` (in module `ironic.drivers.modules.deploy_utils`), 2675
- `parse_instance_info_capabilities()` (in module `ironic.common.utils`), 1358
- `passthru()` (in module `ironic.drivers.base`), 3109
- `PassthruResponse` (class in `ironic.api.controllers.v1.utils`), 931
- `PasswordFileFailedToCreate`, 1170
- `patch()` (in module `ironic.common.args`), 1103
- `patch()` (`ironic.api.controllers.v1.allocation.AllocationsController` method), 809
- `patch()` (`ironic.api.controllers.v1.chassis.ChassisController` method), 817
- `patch()` (`ironic.api.controllers.v1.deploy_template.DeployTemplateController` method), 830
- `patch()` (`ironic.api.controllers.v1.node.NodesController` method), 893
- `patch()` (`ironic.api.controllers.v1.port.PortsController` method), 913
- `patch()` (`ironic.api.controllers.v1.portgroup.PortgroupController` method), 922
- `patch()` (`ironic.api.controllers.v1.volume_connector.VolumeConnectorsController` method), 1032
- `patch()` (`ironic.api.controllers.v1.volume_target.VolumeTargetController` method), 1047
- `patch_update_changed_fields()` (in module `ironic.api.controllers.v1.utils`), 1009
- `patch_validate_allowed_fields()` (in module `ironic.api.controllers.v1.utils`), 1011
- `patched_validate_with_schema()` (in module `ironic.api.controllers.v1.utils`), 1012
- `PatchError`, 1170
- `PathNotFound`, 1170
- `payload` (`ironic.objects.allocation.AllocationCRUDNotification` property), 3166
- `payload` (`ironic.objects.chassis.ChassisCRUDNotification` property), 3207
- `payload` (`ironic.objects.deploy_template.DeployTemplateCRUDNotification` property), 3232
- `payload` (`ironic.objects.node.NodeConsoleNotification` property), 3288
- `payload` (`ironic.objects.node.NodeCorrectedPowerStateNotification` property), 3289
- `payload` (`ironic.objects.node.NodeCRUDNotification` property), 3282
- `payload` (`ironic.objects.node.NodeMaintenanceNotification` property), 3295
- `payload` (`ironic.objects.node.NodeSetPowerStateNotification` property), 3301
- `payload` (`ironic.objects.node.NodeSetProvisionStateNotification` property), 3306
- `payload` (`ironic.objects.port.PortCRUDNotification` property), 3338
- `payload` (`ironic.objects.portgroup.PortgroupCRUDNotification` property), 3355
- `payload` (`ironic.objects.volume_connector.VolumeConnectorCRUDNotification` property), 3384
- `payload` (`ironic.objects.volume_target.VolumeTargetCRUDNotification` property), 3405
- `physical_network`
- `physical_network` (`ironic.db.sqlalchemy.models.Port` attribute), 1777
- `physical_network` (`ironic.objects.port.Port` property), 3333
- `PhysicalNetworkController` (`ironic.objects.port.PortCRUDPayload` property), 3339
- `PHYSNET_PARAM_NAME` (in module `ironic.common.neutron`), 1234
- `PlaybookNotFound`, 1943
- `populate_tenant_network()` (in module `ironic.drivers.modules.network.common`), 347
- `policy_deprecation_check()` (in module `ironic.common.plugins`), 1079
- `pop_node_nested_field()` (in module `ironic.common.utils`), 1360
- `populate_node_uuid()` (in module `ironic.api.controllers.v1.utils`), 1014
- `populate_schema()` (`ironic.objects.notification.NotificationPayloadBase` method), 3316

- populate_storage_driver_internal_info() (in module *ironic.drivers.modules.deploy_utils*), 2677
- Port (class in *ironic.db.sqlalchemy.models*), 1776
- Port (class in *ironic.objects.port*), 3318
- port_changed() (in module *ironic.drivers.base.NetworkInterface* method), 3052
- port_changed() (in module *ironic.drivers.modules.network.common.NeutronClientMixin* method), 2333
- port_changed() (in module *ironic.drivers.modules.network.noop.NoopNetwork* method), 2372
- port_sanitize() (in module *ironic.api.controllers.v1.port*), 915
- PortAlreadyExists, 1170
- PortCRUDNotification (class in *ironic.objects.port*), 3337
- PortCRUDPayload (class in *ironic.objects.port*), 3338
- PortDuplicateName, 1171
- Portgroup (class in *ironic.db.sqlalchemy.models*), 1777
- Portgroup (class in *ironic.objects.portgroup*), 3340
- portgroup_changed() (in module *ironic.drivers.base.NetworkInterface* method), 3053
- portgroup_changed() (in module *ironic.drivers.modules.network.common.NeutronClientMixin* method), 2334
- portgroup_changed() (in module *ironic.drivers.modules.network.noop.NoopNetwork* method), 2373
- portgroup_id (in *ironic.db.sqlalchemy.models.Port* attribute), 1777
- portgroup_id (in *ironic.objects.port.Port* property), 3333
- portgroup_uuid (in *ironic.objects.port.PortCRUDPayload* property), 3339
- PortgroupAlreadyExists, 1171
- PortgroupCRUDNotification (class in *ironic.objects.portgroup*), 3355
- PortgroupCRUDPayload (class in *ironic.objects.portgroup*), 3356
- PortgroupDuplicateName, 1171
- PortgroupMACAlreadyExists, 1171
- PortgroupNotEmpty, 1171
- PortgroupNotFound, 1171
- PortgroupPhysnetInconsistent, 1172
- portgroups (in *ironic.conductor.task_manager.TaskManager* property), 1564
- PortgroupsController (class in *ironic.api.controllers.v1.portgroup*), 916
- PortNotFound, 1171
- ports (in *ironic.conductor.task_manager.TaskManager* property), 1564
- PortsClientMixin (class in *ironic.api.controllers.v1.port*), 905
- post() (in *ironic.api.controllers.v1.allocation.AllocationsController* method), 810
- post() (in *ironic.api.controllers.v1.chassis.ChassisController* method), 818
- post() (in *ironic.api.controllers.v1.deploy_template.DeployTemplate* method), 831
- post() (in *ironic.api.controllers.v1.event.EventsController* method), 842
- post() (in *ironic.api.controllers.v1.node.NodesController* method), 894
- post() (in *ironic.api.controllers.v1.node.NodeVIFController* method), 875
- post() (in *ironic.api.controllers.v1.port.PortsController* method), 914
- post() (in *ironic.api.controllers.v1.portgroup.PortgroupsController* method), 923
- post() (in *ironic.api.controllers.v1.ramdisk.HeartbeatController* method), 924
- post() (in *ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 1076
- post() (in *ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 1051
- post_clean_step_hook() (in module *ironic.drivers.modules.agent_base*), 2586
- post_configuration() (in *ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 2384
- post_create_configuration() (in *ironic.drivers.modules.redfish.raid.RedfishRAID* method), 2464
- post_delete_configuration() (in *ironic.drivers.modules.redfish.raid.RedfishRAID* method), 2466
- post_deploy_step_hook() (in module *ironic.drivers.modules.agent_base*), 2588
- POST_FINISHEDPOST_STATE (in module *ironic.drivers.modules.ilo.common*),

- 2149
- POST_INPOST_STATE (in module *ironic.drivers.modules.ilo.common*), 2149
- POST_INPOSTDISCOVERY_STATE (in module *ironic.drivers.modules.ilo.common*), 2149
- POST_NULL_STATE (in module *ironic.drivers.modules.ilo.common*), 2150
- POST_POWEROFF_STATE (in module *ironic.drivers.modules.ilo.common*), 2150
- post_reset() (*ironic.drivers.modules.redfish.bios.RedfishBios* method), 2385
- POST_RESET_STATE (in module *ironic.drivers.modules.ilo.common*), 2150
- POST_UNKNOWN_STATE (in module *ironic.drivers.modules.ilo.common*), 2150
- POWER (in module *ironic.common.components*), 1129
- power (*ironic.drivers.base.BareDriver* attribute), 2962
- power() (*ironic.api.controllers.v1.node.NodeStatesController* method), 857
- POWER_FAILURE (in module *ironic.common.faults*), 1176
- power_interface (*ironic.db.sqlalchemy.models.Node* attribute), 1774
- power_interface (*ironic.objects.node.Node* property), 3274
- power_interface (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- power_interface (*ironic.objects.node.NodeCRUDPayload* property), 3286
- power_interface (*ironic.objects.node.NodePayload* property), 3299
- power_interface (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- power_interface (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
- POWER_OFF (in module *ironic.common.states*), 1327
- power_off() (*ironic.drivers.modules.agent_client.AgentClient* method), 2619
- power_off() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 2933
- POWER_ON (in module *ironic.common.states*), 1327
- power_on() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 2933
- power_on_node_if_needed() (in module *ironic.conductor.utils*), 1603
- power_reset() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 2934
- power_state (*ironic.db.sqlalchemy.models.Node* attribute), 1774
- power_state (*ironic.objects.node.Node* property), 3275
- power_state (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
- power_state (*ironic.objects.node.NodeCRUDPayload* property), 3286
- power_state (*ironic.objects.node.NodePayload* property), 3299
- power_state (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- power_state (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
- power_state() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 2935
- power_state_error_handler() (in module *ironic.conductor.utils*), 1604
- power_state_for_network_configuration() (in module *ironic.conductor.utils*), 1606
- power_update() (in module *ironic.common.nova*), 1260
- PowerInterface (class in *ironic.drivers.base*), 3064
- PowerStateFailure, 1172
- pre_create_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 2468
- pre_delete_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 2469
- prepare() (*ironic.drivers.base.DeployInterface* method), 2992
- prepare() (*ironic.drivers.modules.agent.CustomAgentDeploy* method), 2545
- prepare() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 1937
- prepare() (*ironic.drivers.modules.fake.FakeDeploy* method), 2722
- prepare() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy*

method), 2903
 prepare() (*ironic.drivers.modules.pxe.PXERamdiskDeploy* (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot*
method), 2910
 prepare_boot_iso() (*in module* *ironic.drivers.modules.image_utils*),
 2798
 prepare_cleaning() (*ironic.drivers.base.DeployInterface*
method), 2993
 prepare_cleaning() (*ironic.drivers.modules.agent_base.AgentBaseMixin* (*ironic.drivers.modules.agent.AgentDeploy*
method), 2555
 prepare_cleaning() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* (*ironic.drivers.modules.agent.CustomAgentDeploy*
method), 1937
 prepare_configdrive_image() (*in module* *ironic.drivers.modules.image_utils*),
 2801
 prepare_deploy_iso() (*in module* *ironic.drivers.modules.image_utils*),
 2803
 prepare_disk_image() (*in module* *ironic.drivers.modules.image_utils*),
 2806
 prepare_floppy_image() (*in module* *ironic.drivers.modules.image_utils*),
 2809
 prepare_host() (*ironic.conductor.base_manager.BaseConductorManager* (*method*), 1382
 prepare_inband_cleaning() (*in module* *ironic.drivers.modules.deploy_utils*),
 2678
 prepare_instance() (*ironic.drivers.base.BootInterface*
method), 2978
 prepare_instance() (*ironic.drivers.modules.fake.FakeBoot*
method), 2707
 prepare_instance() (*ironic.drivers.modules.ilo.boot.IloIPXEBoot*
method), 2142
 prepare_instance() (*ironic.drivers.modules.ilo.boot.IloPXEBoot*
method), 2113
 prepare_instance() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
method), 2119
 prepare_instance() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* (*method*), 2131
 prepare_instance() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot*
method), 2273
 prepare_instance() (*ironic.drivers.modules.pxe_base.PXEBaseMixin*
method), 2917
 prepare_instance() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot*
method), 2394
 prepare_instance_boot() (*ironic.drivers.modules.agent.AgentDeploy*
method), 2523
 prepare_instance_boot() (*ironic.drivers.modules.agent.CustomAgentDeploy*
method), 2548
 prepare_instance_kickstart_config() (*in*
module *ironic.common.pxe_utils*), 1295
 prepare_instance_pxe_config() (*in module*
ironic.common.pxe_utils), 1296
 prepare_instance_to_boot() (*ironic.drivers.modules.agent_base.AgentDeployMixin*
method), 2569
 prepare_node_for_deploy() (*in module*
ironic.drivers.modules.ilo.boot), 2147
 prepare_ramdisk() (*ironic.drivers.base.BootInterface*
method), 2979
 prepare_ramdisk() (*ironic.drivers.modules.fake.FakeBoot*
method), 2708
 prepare_ramdisk() (*ironic.drivers.modules.ilo.boot.IloIPXEBoot*
method), 2143
 prepare_ramdisk() (*ironic.drivers.modules.ilo.boot.IloPXEBoot*
method), 2114
 prepare_ramdisk() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
method), 2122
 prepare_ramdisk() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*
method), 2134
 prepare_ramdisk() (*ironic.drivers.modules.irmc.boot.IRMCIPXEBoot*
method), 2267
 prepare_ramdisk() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot*
method), 2273
 prepare_ramdisk() (*ironic.drivers.modules.pxe_base.PXEBaseMixin*
method), 2917

prepare_ramdisk() (*ironic.drivers.modules.redfish.boot.RedfishBootMixin* property), 3352
 (*ironic.drivers.modules.redfish.boot.RedfishBootMixin* method), 2396
 prepare_service() (*in module properties* (*ironic.objects.volume_target.VolumeTarget* property), 3400
 (*ironic.common.service*), 1315
 previous_provision_state (*properties* (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406
 (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
 (*properties*) (*ironic.api.controllers.v1.driver.DriversController* method), 837
 previous_target_provision_state (*protected* (*ironic.db.sqlalchemy.models.Node* attribute), 1774
 (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3310
 priority (*ironic.db.sqlalchemy.models.DeployTemplateSet* (*ironic.objects.node.Node* property),
 attribute), 1771
 process_event() (*ironic.common.fsm.FSM* protected (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 method), 1182
 process_event() (*protected* (*ironic.objects.node.NodeCRUDPayload*
 (*ironic.conductor.task_manager.TaskManager* property), 3286
 method), 1564
 process_fw_on() (*protected* (*ironic.objects.node.NodePayload*
 (*ironic.drivers.modules.ilo.firmware_process_firmware* property), 3299
 method), 2196
 process_launcher() (*in module protected* (*ironic.objects.node.NodeSetProvisionStatePayload*
 (*ironic.common.service*), 1315
 process_next_step() (*protected_reason*
 (*ironic.drivers.modules.agent_base.AgentDeployMixin* (*ironic.db.sqlalchemy.models.Node*
 method), 2570
 process_next_step() (*protected_reason* (*ironic.objects.node.Node*
 (*ironic.drivers.modules.agent_base.HeartbeatMixin* property), 3275
 method), 2578
 process_next_step() (*protected_reason* (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* property), 3293
 method), 1939
 properties (*ironic.db.sqlalchemy.models.Node* (*ironic.objects.node.NodeCRUDPayload*
 attribute), 1774
 properties (*ironic.db.sqlalchemy.models.Portgroup* protected_reason
 attribute), 1778
 (*ironic.objects.node.NodePayload* prop-
 erty), 3299
 properties (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780
 (*protected_reason*
 (*ironic.objects.node.NodeSetPowerStatePayload*
 property), 3304
 properties (*ironic.objects.node.Node* property), 3275
 (*protected_reason*
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 property), 3293
 (*protected_reason* (*ironic.objects.node.NodeSetProvisionStatePayload*
 property), 3310
 properties (*ironic.objects.node.NodeCRUDPayload* property), 3286
 (*provider* (*ironic.common.dhcp_factory.DHCPFactory*
 property), 1133
 properties (*ironic.objects.node.NodePayload* property), 3299
 (*provision*) (*ironic.api.controllers.v1.node.NodeStatesController*
 method), 860
 properties (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
 (*provision_state*
 (*ironic.objects.node.NodeSetProvisionStatePayload* (*ironic.db.sqlalchemy.models.Node*
 property), 3310
 attribute), 1774
 properties (*ironic.objects.portgroup.Portgroup* provision_state (*ironic.objects.node.Node*

property), 3275

provision_state (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293

provision_state (ironic.objects.node.NodeCRUDPayload property), 3286

provision_state (ironic.objects.node.NodePayload property), 3299

provision_state (ironic.objects.node.NodeSetPowerStatePayload property), 3304

provision_state (ironic.objects.node.NodeSetProvisionStatePayload property), 3310

provision_updated_at (ironic.db.sqlalchemy.models.Node attribute), 1774

provision_updated_at (ironic.objects.node.Node property), 3275

provision_updated_at (ironic.objects.node.NodeCorrectedPowerStatePayload property), 3293

provision_updated_at (ironic.objects.node.NodeCRUDPayload property), 3286

provision_updated_at (ironic.objects.node.NodePayload property), 3299

provision_updated_at (ironic.objects.node.NodeSetPowerStatePayload property), 3304

provision_updated_at (ironic.objects.node.NodeSetProvisionStatePayload property), 3311

provisioning_error_handler() (in module ironic.conductor.utils), 1606

PublicUrlHook (class in ironic.api.hooks), 1076

publish_image() (ironic.drivers.modules.image_utils.ImageHandler method), 2793

publisher (ironic.objects.allocation.AllocationCRUDNotification property), 3166

publisher (ironic.objects.chassis.ChassisCRUDNotification property), 3207

publisher (ironic.objects.deploy_template.DeployTemplateCRUDNotification property), 3232

publisher (ironic.objects.node.NodeConsoleNotification property), 3288

publisher (ironic.objects.node.NodeCorrectedPowerStateNotification property), 3290

publisher (ironic.objects.node.NodeCRUDNotification property), 3282

publisher (ironic.objects.node.NodeMaintenanceNotification property), 3295

publisher (ironic.objects.node.NodeSetPowerStateNotification property), 3301

publisher (ironic.objects.node.NodeSetProvisionStateNotification property), 3306

publisher (ironic.objects.notification.NotificationBase property), 3315

publisher (ironic.objects.port.PortCRUDNotification property), 3338

publisher (ironic.objects.portgroup.PortgroupCRUDNotification property), 3356

publisher (ironic.objects.volume_connector.VolumeConnectorCRUDNotification property), 3384

publisher (ironic.objects.volume_target.VolumeTargetCRUDNotification property), 3405

put() (ironic.api.controllers.v1.node.BootDeviceController method), 844

put() (ironic.api.controllers.v1.node.IndicatorController method), 849

put() (ironic.api.controllers.v1.node.InjectNmiController method), 850

put() (ironic.api.controllers.v1.node.NodeConsoleController method), 854

put() (ironic.api.controllers.v1.node.NodeMaintenanceController method), 855

put() (ironic.api.controllers.v1.node.NodeTraitsController method), 873

PXE (in module ironic.common.boot_devices), 1112

pxe_enabled (ironic.db.sqlalchemy.models.Port attribute), 1777

pxe_enabled (ironic.objects.port.Port property), 3333

pxe_enabled (ironic.objects.port.PortCRUDPayload property), 3339

PXEAnacondaDeploy (class in ironic.drivers.modules.pxe), 2901

PXEBaseMixin (class in ironic.drivers.modules.pxe_base), 2914

PXEBoot (class in ironic.drivers.modules.pxe), 2908

PXERamdiskDeploy (class in ironic.drivers.modules.pxe), 2908

R

raid (ironic.api.controllers.v1.driver.DriversController

attribute), 838
 raid (*ironic.drivers.base.BareDriver* attribute), 2962
 raid() (*ironic.api.controllers.v1.node.NodeStatesController* method), 2083
 method), 870
 RAID_APPLY_CONFIGURATION_ARGSINFO (in module *ironic.drivers.base*), 3083
 RAID_APPLY_CONFIGURATION_ARGSINFO (*ironic.drivers.modules.ibmraid.Ibmraid* attribute), 2089
 raid_config (*ironic.db.sqlalchemy.models.Node* attribute), 1774
 raid_config (*ironic.objects.node.Node* property), 3275
 raid_interface (*ironic.db.sqlalchemy.models.Node* attribute), 1774
 raid_interface (*ironic.objects.node.Node* property), 3275
 raid_interface (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3293
 raid_interface (*ironic.objects.node.NodeCRUDPayload* property), 3286
 raid_interface (*ironic.objects.node.NodePayload* property), 3299
 raid_interface (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
 raid_interface (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3311
 RAIDInterface (class in *ironic.drivers.base*), 3071
 ramdisk_ref (*ironic.objects.deployment.Deployment* property), 3241
 read_only (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
 read_only (*ironic.objects.bios.BIOSSetting* property), 3188
 REBOOT (in module *ironic.common.states*), 1327
 reboot() (*ironic.drivers.base.PowerInterface* method), 3066
 reboot() (*ironic.drivers.modules.agent_client.AgentClient* method), 2621
 reboot() (*ironic.drivers.modules.agent_power.AgentPower* method), 2629
 reboot() (*ironic.drivers.modules.drac.power.DracPower* method), 2011
 reboot() (*ironic.drivers.modules.fake.FakePower* method), 2760
 reboot() (*ironic.drivers.modules.ibmraid.power.IBMCPower* method), 2242
 reboot() (*ironic.drivers.modules.ilo.power.IloPower* method), 2242
 reboot() (*ironic.drivers.modules.ipmitool.IPMIPower* method), 2835
 reboot() (*ironic.drivers.modules.irmc.power.IRMCPower* method), 2319
 reboot() (*ironic.drivers.modules.redfish.power.RedfishPower* method), 2452
 reboot() (*ironic.drivers.modules.snmp.SNMPower* method), 2945
 reboot() (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 2520
 reboot_and_finish_deploy() (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2571
 reboot_to_instance() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 2579
 reboot_to_instance() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy* method), 2905
 REBUILD (in module *ironic.common.states*), 1328
 RedfishBIOS (class in *ironic.drivers.modules.redfish.bios*), 2379
 RedfishConnectionError, 1172
 RedfishError, 1172
 RedfishHardware (class in *ironic.drivers.modules.redfish*), 3135
 RedfishInspect (class in *ironic.drivers.modules.redfish.inspect*), 2403
 RedfishManagement (class in *ironic.drivers.modules.redfish.management*), 2408
 RedfishPower (class in *ironic.drivers.modules.redfish.power*), 2448
 RedfishRAID (class in *ironic.drivers.modules.redfish.raid*), 2457
 RedfishVendorPassthru (class in *ironic.drivers.modules.redfish.vendor*), 2486
 RedfishVirtualMediaBoot (class in *ironic.drivers.modules.redfish.boot*), 2388

[refresh\(\)](#) (*ironic.objects.allocation.Allocation* method), 3162
[refresh\(\)](#) (*ironic.objects.chassis.Chassis* method), 3204
[refresh\(\)](#) (*ironic.objects.conductor.Conductor* method), 3212
[refresh\(\)](#) (*ironic.objects.deploy_template.DeployTemplate* method), 3228
[refresh\(\)](#) (*ironic.objects.deployment.Deployment* method), 3241
[refresh\(\)](#) (*ironic.objects.node.Node* method), 3275
[refresh\(\)](#) (*ironic.objects.port.Port* method), 3333
[refresh\(\)](#) (*ironic.objects.portgroup.Portgroup* method), 3352
[refresh\(\)](#) (*ironic.objects.volume_connector.VolumeConnector* method), 3380
[refresh\(\)](#) (*ironic.objects.volume_target.VolumeTarget* method), 3400
[refresh_clean_steps\(\)](#) (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 2579
[refresh_steps\(\)](#) (*ironic.drivers.modules.agent_base.AgentBaseMixin* method), 2556
[refresh_steps\(\)](#) (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 2573
[refresh_steps\(\)](#) (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 2580
[register\(\)](#) (*ironic.objects.conductor.Conductor* class method), 3213
[register_all\(\)](#) (in module *ironic.objects*), 3406
[register_auth_opts\(\)](#) (in module *ironic.conf.auth*), 1622
[register_conductor\(\)](#) (*ironic.db.api.Connection* method), 1875
[register_conductor\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1722
[register_conductor_hardware_interfaces\(\)](#) (*ironic.db.api.Connection* method), 1878
[register_conductor_hardware_interfaces\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 1724
[register_hardware_interfaces\(\)](#) (*ironic.objects.conductor.Conductor* method), 3215
[register_opts\(\)](#) (in module *ironic.conf.agent*), 1620
[register_opts\(\)](#) (in module *ironic.conf.anaconda*), 1621
[register_opts\(\)](#) (in module *ironic.conf.ansible*), 1621
[register_opts\(\)](#) (in module *ironic.conf.api*), 1621
[register_opts\(\)](#) (in module *ironic.conf.audit*), 1621
[register_opts\(\)](#) (in module *ironic.conf.cinder*), 1622
[register_opts\(\)](#) (in module *ironic.conf.conductor*), 1623
[register_opts\(\)](#) (in module *ironic.conf.console*), 1623
[register_opts\(\)](#) (in module *ironic.conf.database*), 1623
[register_opts\(\)](#) (in module *ironic.conf.default*), 1623
[register_opts\(\)](#) (in module *ironic.conf.deploy*), 1623
[register_opts\(\)](#) (in module *ironic.conf.dhcp*), 1623
[register_opts\(\)](#) (in module *ironic.conf.drac*), 1623
[register_opts\(\)](#) (in module *ironic.conf.glance*), 1623
[register_opts\(\)](#) (in module *ironic.conf.healthcheck*), 1624
[register_opts\(\)](#) (in module *ironic.conf.ibmcc*), 1624
[register_opts\(\)](#) (in module *ironic.conf.ilo*), 1624
[register_opts\(\)](#) (in module *ironic.conf.inspector*), 1624
[register_opts\(\)](#) (in module *ironic.conf.ipmi*), 1624
[register_opts\(\)](#) (in module *ironic.conf.irmc*), 1624
[register_opts\(\)](#) (in module *ironic.conf.metrics*), 1624
[register_opts\(\)](#) (in module *ironic.conf.metrics_statsd*), 1624
[register_opts\(\)](#) (in module *ironic.conf.molds*), 1625
[register_opts\(\)](#) (in module *ironic.conf.neutron*), 1625
[register_opts\(\)](#) (in module *ironic.conf.nova*), 1625
[register_opts\(\)](#) (in module *ironic.conf.pxe*), 1625

- 1627
- `register_opts()` (in module `ironic.conf.redfish`), 1627
- `register_opts()` (in module `ironic.conf.service_catalog`), 1627
- `register_opts()` (in module `ironic.conf.snmp`), 1627
- `register_opts()` (in module `ironic.conf.swift`), 1627
- `register_opts()` (in module `ironic.conf.xclarity`), 1628
- `registration_hook()` (`ironic.objects.base.IronicObjectRegistry` method), 3179
- `registry_fields` (`ironic.objects.bios.BIOSSetting` attribute), 3188
- `reject_fields_in_newer_versions()` (in module `ironic.api.controllers.v1.node`), 899
- `reject_patch_in_newer_versions()` (in module `ironic.api.controllers.v1.node`), 899
- `release()` (`ironic.objects.node.Node` class method), 3276
- `release_node()` (`ironic.db.api.Connection` method), 1880
- `release_node()` (`ironic.db.sqlalchemy.api.Connection` method), 1727
- `release_port()` (in module `ironic.drivers.modules.console_utils`), 2645
- `release_resources()` (`ironic.conductor.task_manager.TaskManager` method), 1566
- `remove()` (`ironic.drivers.modules.ilo.firmware_processor.FirmwareProcessor` method), 2195
- `remove_agent_url()` (in module `ironic.conductor.utils`), 1608
- `remove_cleaning_network()` (`ironic.drivers.base.NetworkInterface` method), 3053
- `remove_cleaning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 2353
- `remove_cleaning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 2363
- `remove_cleaning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 2374
- `remove_http_instance_symlink()` (in module `ironic.drivers.modules.deploy_utils`), 2680
- `remove_image_from_swift()` (in module `ironic.drivers.modules.ilo.common`), 2172
- `remove_image_from_web_server()` (in module `ironic.drivers.modules.ilo.common`), 2173
- `remove_inspection_network()` (`ironic.drivers.base.NetworkInterface` method), 3054
- `remove_inspection_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 2354
- `remove_inspection_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 2364
- `remove_neutron_ports()` (in module `ironic.common.neutron`), 1247
- `remove_node_rescue_password()` (in module `ironic.conductor.utils`), 1608
- `remove_node_traits()` (`ironic.conductor.manager.ConductorManager` method), 1407
- `remove_node_traits()` (`ironic.conductor.rpcapi.ConductorAPI` method), 1511
- `remove_ports_from_network()` (in module `ironic.common.neutron`), 1248
- `remove_provisioning_network()` (`ironic.drivers.base.NetworkInterface` method), 3055
- `remove_provisioning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 2355
- `remove_provisioning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 2365
- `remove_provisioning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 2374
- `remove_rescuing_network()` (`ironic.drivers.base.NetworkInterface` method), 3056
- `remove_rescuing_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 2355
- `remove_rescuing_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 2365

- method*), 2365
- `remove_single_or_list_of_files()` (in module *ironic.drivers.modules.ilo.common*), 2174
- `remove_vifs_from_node()` (in module *ironic.common.network*), 1232
- `render_template()` (in module *ironic.common.utils*), 1361
- `replace_node_id_with_uuid()` (in module *ironic.api.controllers.v1.utils*), 1016
- `replace_node_uuid_with_id()` (in module *ironic.api.controllers.v1.utils*), 1017
- `RequestContext` (class in *ironic.common.context*), 1130
- `RequestContextSerializer` (class in *ironic.common.rpc*), 1309
- `require_exclusive_lock()` (in module *ironic.conductor.task_manager*), 1572
- `RESCUE` (in module *ironic.common.states*), 1328
- `rescue` (*ironic.drivers.base.BareDriver* attribute), 2963
- `rescue()` (*ironic.drivers.base.RescueInterface* method), 3085
- `rescue()` (*ironic.drivers.modules.agent.AgentRescue* method), 2535
- `rescue()` (*ironic.drivers.modules.fake.FakeRescue* method), 2771
- `rescue()` (*ironic.drivers.modules.noop.NoRescue* method), 2882
- `RESCUE_ABORT_FAILURE` (in module *ironic.common.faults*), 1177
- `rescue_interface` (*ironic.db.sqlalchemy.models.Node* attribute), 1774
- `rescue_interface` (*ironic.objects.node.Node* property), 3277
- `rescue_interface` (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3294
- `rescue_interface` (*ironic.objects.node.NodeCRUDPayload* property), 3287
- `rescue_interface` (*ironic.objects.node.NodePayload* property), 3299
- `rescue_interface` (*ironic.objects.node.NodeSetPowerStatePayload* property), 3304
- `rescue_interface` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3311
- `rescue_or_deploy_mode()` (in module *ironic.drivers.modules.deploy_utils*), 2680
- `RESCUEFAIL` (in module *ironic.common.states*), 1328
- `RescueInterface` (class in *ironic.drivers.base*), 3083
- `RESCUEWAIT` (in module *ironic.common.states*), 1328
- `RESCUING` (in module *ironic.common.states*), 1329
- `rescuing_error_handler()` (in module *ironic.conductor.utils*), 1609
- `reservation` (*ironic.db.sqlalchemy.models.Node* attribute), 1774
- `reservation` (*ironic.objects.node.Node* property), 3277
- `reserve()` (*ironic.objects.node.Node* class method), 3277
- `reserve_node()` (*ironic.db.api.Connection* method), 1881
- `reserve_node()` (*ironic.db.sqlalchemy.api.Connection* method), 1728
- `reset()` (*ironic.common.hash_ring.HashRingManager* class method), 1183
- `reset()` (*ironic.common.wsgi_service.WSGIService* method), 1371
- `reset_bios_to_default()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2222
- `reset_idrac()` (*ironic.drivers.modules.drac.management.DracRedfishManager* method), 1994
- `reset_idrac()` (*ironic.drivers.modules.drac.management.DracRedfishManager* method), 2001
- `reset_ilo()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2224
- `reset_ilo_credential()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2225
- `reset_required` (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
- `reset_required` (*ironic.objects.bios.BIOSSetting* property), 3188
- `reset_secure_boot_keys_to_default()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 2226
- `reset_secure_boot_keys_to_default()` (*ironic.drivers.modules.redfish.management.RedfishManager* method), 2226

method), 2434
resolve_type()
(ironic.api.functions.FunctionArgument method), 1065
resolve_types()
(ironic.api.functions.FunctionDefinition method), 1067
resource_class
(ironic.db.sqlalchemy.models.Allocation attribute), 1766
resource_class
(ironic.db.sqlalchemy.models.Node attribute), 1774
resource_class
(ironic.objects.allocation.Allocation property), 3164
resource_class
(ironic.objects.allocation.AllocationCRUDPayload property), 3287
resource_class (*ironic.objects.node.Node property*), 3278
resource_class
(ironic.objects.node.NodeCorrectedPowerStatePayload property), 3294
resource_class
(ironic.objects.node.NodeCRUDPayload property), 3287
resource_class
(ironic.objects.node.NodePayload property), 3300
resource_class
(ironic.objects.node.NodeSetPowerStatePayload property), 3305
resource_class
(ironic.objects.node.NodeSetProvisionStatePayload property), 3311
restore_boot_device()
(ironic.drivers.modules.redfish.management.RedfishManagement method), 2435
restore_irmc_bios_config()
(ironic.drivers.modules.irmc.management.IRMCMgmt command line option method), 2308
restore_power_state_if_needed() (*in module ironic.conductor.utils*), 1611
retired (*ironic.db.sqlalchemy.models.Node attribute*), 1774
retired (*ironic.objects.node.Node property*), 3278
retired (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 3294
retired (*ironic.objects.node.NodeCRUDPayload property*), 3287
retired (*ironic.objects.node.NodePayload property*), 3300
retired (*ironic.objects.node.NodeSetPowerStatePayload property*), 3305
retired (*ironic.objects.node.NodeSetProvisionStatePayload property*), 3311
retry_interval
(ironic.drivers.modules.snmp.SNMPDriverBase attribute), 2935
retry_on_outdated_cache() (*in module ironic.drivers.modules.snmp*), 2950
return_type (*ironic.api.functions.FunctionDefinition attribute*), 1067
revert_dictionary() (*in module ironic.drivers.modules.ibmcmdutils*), 2099
RedfishManagement
ironic-dbsync command line option, 537
ironic-dbsync command line option
--autogenerate, 539
--help, 539
--message <MESSAGE>, 539
-h, 539
-m <MESSAGE>, 539
revision() (*in module ironic.db.migration*), 1911
revision() (*in module ironic.db.sqlalchemy.migration*), 1763
revision() (*ironic.cmd.dbsync.DBCommand*

method), 1083

ring (*ironic.common.hash_ring.HashRingManager* property), 1183

rmtree_without_raise() (in module *ironic.common.utils*), 1363

rollback_ports() (in module *ironic.common.neutron*), 1249

root() (in module *ironic.api.controllers.root*), 1056

root_device (*ironic.objects.deployment.Deployment* property), 3241

root_gib (*ironic.objects.deployment.Deployment* property), 3241

RootController (class in *ironic.api.controllers.root*), 1056

RPC_API_VERSION (*ironic.conductor.manager.ConductorManager* attribute), 1396

RPC_API_VERSION (*ironic.conductor.rpcapi.ConductorAPI* attribute), 1430

RPCHook (class in *ironic.api.hooks*), 1078

RPCService (class in *ironic.common.rpc_service*), 1313

S

SAFE (in module *ironic.common.boot_devices*), 1112

safe_rstrip() (in module *ironic.common.utils*), 1363

sanitize_dict() (in module *ironic.api.controllers.v1.utils*), 1018

save() (*ironic.objects.allocation.Allocation* method), 3164

save() (*ironic.objects.bios.BIOSSetting* method), 3188

save() (*ironic.objects.bios.BIOSSettingList* class method), 3194

save() (*ironic.objects.chassis.Chassis* method), 3205

save() (*ironic.objects.conductor.Conductor* method), 3216

save() (*ironic.objects.deploy_template.DeployTemplate* method), 3229

save() (*ironic.objects.node.Node* method), 3279

save() (*ironic.objects.port.Port* method), 3334

save() (*ironic.objects.portgroup.Portgroup* method), 3353

save() (*ironic.objects.volume_connector.VolumeConnector* method), 3381

save() (*ironic.objects.volume_target.VolumeTarget* method), 3402

save_configuration() (in module *ironic.common.molds*), 1224

SCHEMA (*ironic.objects.allocation.AllocationCRUDPayload* attribute), 3167

SCHEMA (*ironic.objects.chassis.ChassisCRUDPayload* attribute), 3207

SCHEMA (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 3232

SCHEMA (*ironic.objects.node.NodeCRUDPayload* attribute), 3282

SCHEMA (*ironic.objects.node.NodePayload* attribute), 3296

SCHEMA (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 3307

SCHEMA (*ironic.objects.notification.NotificationPayloadBase* attribute), 3315

SCHEMA (*ironic.objects.port.PortCRUDPayload* attribute), 3338

SCHEMA (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 3356

SCHEMA (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 3385

SCHEMA (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 3405

schema() (in module *ironic.common.args*), 1103

second_method() (*ironic.drivers.modules.fake.FakeVendorB* method), 2785

security_parameters_update() (*ironic.drivers.modules.ilo.management.IloManagement* method), 2227

send_raw() (in module *ironic.drivers.modules.ipmitool*), 2858

send_raw() (*ironic.drivers.modules.ipmitool.VendorPassthru* method), 2850

serialize_context() (*ironic.common.rpc.RequestContextSerializer* method), 1311

serialize_entity() (*ironic.common.rpc.RequestContextSerializer* method), 1312

serialize_entity() (*ironic.objects.base.IronicObjectSerializer* method), 3179

service (*ironic.objects.notification.NotificationPublisher* property), 3317

ServiceUnavailable, 1172

SessionCache (class in *ironic.drivers.modules.redfish.utils*), 2479

[set\(\)](#) ([ironic.drivers.modules.snmp.SNMPClient](#) method), 2927
[set_arg_types\(\)](#) ([ironic.api.functions.FunctionDefinition](#) method), 1067
[set_async_step_flags\(\)](#) (in module [ironic.drivers.modules.deploy_utils](#)), 2680
[set_bios_config\(\)](#) ([ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthru](#) method), 2060
[set_boot_device\(\)](#) (in module [ironic.drivers.modules.drac.management](#)), 2005
[set_boot_device\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 1407
[set_boot_device\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 1513
[set_boot_device\(\)](#) ([ironic.drivers.base.ManagementInterface](#) method), 3033
[set_boot_device\(\)](#) ([ironic.drivers.modules.drac.management.DracManagement](#) method), 2002
[set_boot_device\(\)](#) ([ironic.drivers.modules.fake.FakeManagement](#) method), 2753
[set_boot_device\(\)](#) ([ironic.drivers.modules.ibmcm.management.IBMCManagement](#) method), 2074
[set_boot_device\(\)](#) ([ironic.drivers.modules.ilo.management.IloManagement](#) method), 2228
[set_boot_device\(\)](#) ([ironic.drivers.modules.ipmitool.IPMIManagement](#) method), 2828
[set_boot_device\(\)](#) ([ironic.drivers.modules.irmc.management.IBMCManagement](#) method), 2309
[set_boot_device\(\)](#) ([ironic.drivers.modules.noop_mgmt.NoopManagement](#) method), 2897
[set_boot_device\(\)](#) ([ironic.drivers.modules.redfish.management.RedfishManagement](#) method), 2437
[set_boot_device\(\)](#) ([ironic.drivers.modules.xclarity.management.XclarityManagement](#) method), 2515
[set_boot_mode\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 2175
[set_boot_mode\(\)](#) ([ironic.drivers.base.ManagementInterface](#) method), 3035
[set_boot_mode\(\)](#) ([ironic.drivers.modules.ibmcm.management.IBMCManagement](#) method), 2076
[set_boot_mode\(\)](#) ([ironic.drivers.modules.ilo.management.IloManagement](#) method), 2230
[set_boot_mode\(\)](#) ([ironic.drivers.modules.redfish.management.RedfishManagement](#) method), 2439
[set_config\(\)](#) (in module [ironic.drivers.modules.drac.bios](#)), 1965
[set_console_mode\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 1408
[set_console_mode\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 1516
[set_defaults\(\)](#) (in module [ironic.common.rpc](#)), 1313
[set_defaults\(\)](#) ([ironic.drivers.modules.deploy_utils](#)), 2682
[set_indicator_state\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 1408
[set_indicator_state\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 1518
[set_indicator_state\(\)](#) ([ironic.drivers.base.ManagementInterface](#) method), 3038
[set_indicator_state\(\)](#) ([ironic.drivers.modules.redfish.management.RedfishManagement](#) method), 2441
[set_boot_target\(\)](#) ([ironic.drivers.modules.ilo.management.IloManagement](#) method), 2232
[set_node_cleaning_steps\(\)](#) (in module [ironic.conductor.steps](#)), 1549
[set_node_deployment_steps\(\)](#) (in module [ironic.conductor.steps](#)), 1550
[set_node_nested_field\(\)](#) (in module [ironic.common.utils](#)), 1364
[set_node_tags\(\)](#) ([ironic.db.api.Connection](#) method), 1883
[set_node_tags\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 2175

(ironic.db.sqlalchemy.api.Connection method), 1730
set_node_traits() (*ironic.db.api.Connection method*), 1884
set_node_traits() (*ironic.db.sqlalchemy.api.Connection method*), 1731
set_options() (*ironic.api.functions.FunctionDefinition method*), 1067
set_power_state() (*ironic.drivers.base.PowerInterface method*), 3068
set_power_state() (*ironic.drivers.modules.agent_power.AgentPower method*), 2630
set_power_state() (*ironic.drivers.modules.drac.power.DracWSManPower method*), 2013
set_power_state() (*ironic.drivers.modules.fake.FakePower method*), 2762
set_power_state() (*ironic.drivers.modules.ibmcpower.IBMCPower method*), 2085
set_power_state() (*ironic.drivers.modules.ilo.power.IloPower method*), 2244
set_power_state() (*ironic.drivers.modules.ipmitool.IPMIPower method*), 2837
set_power_state() (*ironic.drivers.modules.irmc.power.IRMCPower method*), 2321
set_power_state() (*ironic.drivers.modules.redfish.power.RedfishPower method*), 2454
set_power_state() (*ironic.drivers.modules.snmp.SNMPPower method*), 2947
set_power_state() (*ironic.drivers.modules.xclarity.power.XClarityPower method*), 2520
set_raid_settings() (in module *ironic.drivers.modules.drac.raid*), 2045
set_secure_boot_mode() (in module *ironic.drivers.modules.ilo.common*), 2176
set_secure_boot_mode() (in module *ironic.drivers.modules.irmc.common*), 2290
set_secure_boot_state() (*ironic.drivers.base.ManagementInterface method*), 3040
set_secure_boot_state() (*ironic.drivers.modules.ilo.management.IloManagement method*), 2233
set_secure_boot_state() (*ironic.drivers.modules.irmc.management.IRMCMangement method*), 2311
set_secure_boot_state() (*ironic.drivers.modules.redfish.management.RedfishManagement method*), 2444
set_spawn_error_hook() (*ironic.conductor.task_manager.TaskManager method*), 1567
set_target_raid_config() (*ironic.conductor.manager.ConductorManager method*), 1408
set_target_raid_config() (*ironic.conductor.rpcapi.ConductorAPI method*), 1521
setup() (in module *ironic.common.profiler*), 1266
setup_app() (in module *ironic.api.app*), 1064
setup_uefi_https() (in module *ironic.drivers.modules.ilo.common*), 2177
setup_vmedia() (in module *ironic.drivers.modules.ilo.common*), 2180
setup_vmedia_for_boot() (in module *ironic.drivers.modules.ilo.common*), 2183
should_manage_boot() (*ironic.drivers.modules.agent.CustomAgentDeploy method*), 2548
should_manage_boot() (*ironic.drivers.modules.agent_base.AgentBaseMixin method*), 2557
should_manage_boot() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy method*), 2905
should_write_image() (*ironic.drivers.base.StorageInterface method*), 3089
should_write_image() (*ironic.drivers.modules.fake.FakeStorage method*), 2778
should_write_image() (*ironic.drivers.modules.storage.cinder.CinderStorage method*), 2494
should_write_image()

(*ironic.drivers.modules.storage.external.ExternalStorage* method), 2498

`should_write_image()` (*ironic.drivers.modules.storage.noop.NoopStorage* method), 2504

`show()` (*ironic.common.glance_service.image_service.GlanceImageService* method), 1086

`show()` (*ironic.common.image_service.BaseImageService* method), 1185

`show()` (*ironic.common.image_service.FileImageService* method), 1188

`show()` (*ironic.common.image_service.HttpImageService* method), 1192

`sig` (in module *ironic.api.functions*), 1068

`signature` (class in *ironic.api.functions*), 1068

`simple_headers` (*ironic.api.app.IronicCORS* attribute), 1064

`skip_automated_cleaning()` (in module *ironic.conductor.utils*), 1611

`SNMPClient` (class in *ironic.drivers.modules.snmp*), 2924

`SNMPDriverAPCMasterSwitch` (class in *ironic.drivers.modules.snmp*), 2928

`SNMPDriverAPCMasterSwitchPlus` (class in *ironic.drivers.modules.snmp*), 2929

`SNMPDriverAPCRackPDU` (class in *ironic.drivers.modules.snmp*), 2930

`SNMPDriverAten` (class in *ironic.drivers.modules.snmp*), 2931

`SNMPDriverAuto` (class in *ironic.drivers.modules.snmp*), 2932

`SNMPDriverBase` (class in *ironic.drivers.modules.snmp*), 2932

`SNMPDriverBaytechMRP27` (class in *ironic.drivers.modules.snmp*), 2935

`SNMPDriverCyberPower` (class in *ironic.drivers.modules.snmp*), 2936

`SNMPDriverEatonPower` (class in *ironic.drivers.modules.snmp*), 2937

`SNMPDriverSimple` (class in *ironic.drivers.modules.snmp*), 2939

`SNMPDriverTeltronix` (class in *ironic.drivers.modules.snmp*), 2941

`SNMPFailure`, 1172

`SNMPHardware` (class in *ironic.drivers.snmp*), 3137

`SNMPPower` (class in *ironic.drivers.modules.snmp*), 2942

`SOFT_POWER_OFF` (in module *ironic.common.states*), 1329

`SOFT_REBOOT` (in module *ironic.common.states*), 1330

`spawn_after()` (*ironic.conductor.task_manager.TaskManager* method), 1569

`spawn_cleaning_error_handler()` (in module *ironic.conductor.utils*), 1612

`spawn_deploy_error_handler()` (in module *ironic.conductor.utils*), 1612

`spawn_rescue_error_handler()` (in module *ironic.conductor.utils*), 1612

`STABLE_STATES` (in module *ironic.common.states*), 1330

`stamp`

`stamp` (*ironic-dbsync* command line option), 537

`stamp` command line option

`--help`, 539

`--revision <REVISION>`, 539

`-h`, 539

`stamp()` (in module *ironic.db.migration*), 1911

`stamp()` (in module *ironic.db.sqlalchemy.migration*), 1763

`stamp()` (*ironic.cmd.dbsync.DBCommand* method), 1083

`standalone_ports_supported` (*ironic.db.sqlalchemy.models.Portgroup* attribute), 1778

`standalone_ports_supported` (*ironic.objects.portgroup.Portgroup* property), 3354

`standalone_ports_supported` (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 3357

`START` (*ironic.objects.fields.NotificationStatus* attribute), 3247

`start()` (*ironic.common.rpc_service.RPCService* method), 1314

`start()` (*ironic.common.wsgi_service.WSGIService* method), 1371

`start_console()` (*ironic.drivers.base.ConsoleInterface* method), 2985

`start_console()` (*ironic.drivers.modules.fake.FakeConsole* method), 2715

`start_console()` (*ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole* method), 2842

`start_console()` (*ironic.drivers.modules.ipmitool.IPMISocatConsole* method), 2845

`start_console()`

(ironic.drivers.modules.noop.NoConsole method), 2872
start_deploy() (in module *ironic.conductor.deployments*), 1390
start_shellinabox_console() (in module *ironic.drivers.modules.console_utils*), 2645
start_socat_console() (in module *ironic.drivers.modules.console_utils*), 2647
state (*ironic.db.sqlalchemy.models.Allocation* attribute), 1766
state (*ironic.objects.allocation.Allocation* property), 3165
state (*ironic.objects.allocation.AllocationCRUDPayload* property), 3168
state (*ironic.objects.deployment.Deployment* property), 3241
states (*ironic.api.controllers.v1.node.NodesController* attribute), 895
status (*ironic.objects.notification.EventType* property), 3312
status_code (*ironic.api.controllers.v1.utils.PassthruResponse* attribute), 931
status_code (*ironic.api.functions.FunctionDefinition* attribute), 1067
status_off (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 2939
status_on (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 2939
status_pending_off (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 2939
status_pending_on (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 2939
step (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 1771
step_sanitize() (in module *ironic.api.controllers.v1.deploy_template*), 832
steps (*ironic.objects.deploy_template.DeployTemplateStep* property), 3231
steps (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* property), 3233
stop() (*ironic.common.rpc_service.RPCService* method), 1314
stop() (*ironic.common.wsgi_service.WSGIService* method), 1371
stop_console() (*ironic.drivers.base.ConsoleInterface* method), 2986
stop_console() (*ironic.drivers.modules.fake.FakeConsole* method), 2715
stop_console() (*ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole* method), 2844
stop_console() (*ironic.drivers.modules.ipmitool.IPMISocatConsole* method), 2847
stop_console() (*ironic.drivers.modules.noop.NoConsole* method), 2873
stop_shellinabox_console() (in module *ironic.drivers.modules.console_utils*), 2648
stop_socat_console() (in module *ironic.drivers.modules.console_utils*), 2649
storage (*ironic.drivers.base.BareDriver* attribute), 2963
storage_interface (*ironic.db.sqlalchemy.models.Node* attribute), 1774
storage_interface (*ironic.objects.node.Node* property), 3280
StorageInterface (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3294
storage_interface (*ironic.objects.node.NodeCRUDPayload* property), 3287
storage_interface (*ironic.objects.node.NodePayload* property), 3300
storage_interface (*ironic.objects.node.NodeSetPowerStatePayload* property), 3305
storage_interface (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3311
StorageError, 1172
StorageInterface (class in *ironic.drivers.base*), 1137
StorageInterfaceFactory (class in *ironic.common.driver_factory*), 1137
store_agent_certificate() (in module *ironic.conductor.utils*), 1612
store_ramdisk_logs() (in module *ironic.drivers.utils*), 3150
string (*ironic.api.controllers.base.Version*

- attribute*), 1056
- `string()` (in module *ironic.common.args*), 1104
- `string_list()` (in module *ironic.common.args*), 1105
- `StringAcceptsCallable` (class in *ironic.objects.fields*), 3248
- `StringField` (class in *ironic.objects.fields*), 3249
- `StringFieldThatAcceptsCallable` (class in *ironic.objects.fields*), 3249
- `STUCK_STATES_TREATED_AS_FAIL` (in module *ironic.common.states*), 1330
- `SUCCESS` (*ironic.objects.fields.NotificationStatus* attribute), 3247
- `supported` (*ironic.drivers.base.BaseInterface* attribute), 2973
- `supported` (*ironic.drivers.hardware_type.AbstractHardwareType* attribute), 3122
- `supported()` (*ironic.api.controllers.v1.node.BootDeviceController* method), 846
- `supported_bios_interfaces` (*ironic.drivers.drac.IDRACHardware* property), 3110
- `supported_bios_interfaces` (*ironic.drivers.fake_hardware.FakeHardware* property), 3113
- `supported_bios_interfaces` (*ironic.drivers.hardware_type.AbstractHardwareType* property), 3122
- `supported_bios_interfaces` (*ironic.drivers.hardware_type.AbstractHardwareType* property), 3122
- `supported_bios_interfaces` (*ironic.drivers.ilo.IloHardware* property), 3128
- `supported_bios_interfaces` (*ironic.drivers.irmc.IRMCHardware* property), 3133
- `supported_bios_interfaces` (*ironic.drivers.redfish.RedfishHardware* property), 3135
- `supported_boot_interfaces` (*ironic.drivers.drac.IDRACHardware* property), 3111
- `supported_boot_interfaces` (*ironic.drivers.fake_hardware.FakeHardware* property), 3113
- `supported_boot_interfaces` (*ironic.drivers.generic.GenericHardware* property), 3117
- `supported_boot_interfaces` (*ironic.drivers.hardware_type.AbstractHardwareType* property), 3122
- `supported_boot_interfaces` (*ironic.drivers.ilo.Ilo5Hardware* property), 3127
- `supported_boot_interfaces` (*ironic.drivers.ilo.IloHardware* property), 3128
- `supported_boot_interfaces` (*ironic.drivers.irmc.IRMCHardware* property), 3134
- `supported_boot_interfaces` (*ironic.drivers.redfish.RedfishHardware* property), 3135
- `SUPPORTED_BOOT_MODE_LEGACY_BIOS_AND_UEFI` (in module *ironic.drivers.modules.ilo.common*), 2150
- `SUPPORTED_BOOT_MODE_LEGACY_BIOS_ONLY` (in module *ironic.drivers.modules.ilo.common*), 2151
- `SUPPORTED_BOOT_MODE_UEFI_ONLY` (in module *ironic.drivers.modules.ilo.common*), 2151
- `supported_console_interfaces` (*ironic.drivers.fake_hardware.FakeHardware* property), 3113
- `supported_console_interfaces` (*ironic.drivers.hardware_type.AbstractHardwareType* property), 3122
- `supported_console_interfaces` (*ironic.drivers.ilo.IloHardware* property), 3128
- `supported_console_interfaces` (*ironic.drivers.ipmi.IPMIHardware* property), 3131
- `supported_console_interfaces` (*ironic.drivers.irmc.IRMCHardware* property), 3134
- `supported_deploy_interfaces` (*ironic.drivers.fake_hardware.FakeHardware* property), 3114
- `supported_deploy_interfaces` (*ironic.drivers.generic.GenericHardware* property), 3117
- `supported_deploy_interfaces` (*ironic.drivers.hardware_type.AbstractHardwareType* property), 3123
- `supported_inspect_interfaces` (*ironic.drivers.drac.IDRACHardware* property), 3111
- `supported_inspect_interfaces` (*ironic.drivers.fake_hardware.FakeHardware* property), 3114

- supported_inspect_interfaces
(*ironic.drivers.generic.GenericHardware* property), 3117
- supported_inspect_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* & *ironic.drivers.fake_hardware.FakeHardware* property), 3123
- supported_inspect_interfaces
(*ironic.drivers.ilo.IloHardware* property), 3129
- supported_inspect_interfaces
(*ironic.drivers.irmc.IRMCHardware* property), 3134
- supported_inspect_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3135
- supported_management_interfaces
(*ironic.drivers.drac.IDRACHardware* property), 3111
- supported_management_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3114
- supported_management_interfaces
(*ironic.drivers.generic.ManualManagementHardware* property), 3119
- supported_management_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* & *ironic.drivers.ibm.ibmchardware* property), 3123
- supported_management_interfaces
(*ironic.drivers.ibm.ibmchardware* property), 3125
- supported_management_interfaces
(*ironic.drivers.ilo.Ilo5Hardware* property), 3127
- supported_management_interfaces
(*ironic.drivers.ilo.IloHardware* property), 3129
- supported_management_interfaces
(*ironic.drivers.intel_ipmi.IntelIPMIHardware* property), 3130
- supported_management_interfaces
(*ironic.drivers.ipmi.IPMIHardware* property), 3132
- supported_management_interfaces
(*ironic.drivers.irmc.IRMCHardware* property), 3134
- supported_management_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3136
- supported_management_interfaces
(*ironic.drivers.snmp.SNMPHardware* property), 3137
- supported_management_interfaces
(*ironic.drivers.xclarity.XClarityHardware* property), 3153
- supported_network_interfaces
(*ironic.drivers.generic.GenericHardware* property), 3117
- supported_network_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3123
- supported_network_interfaces
(*ironic.drivers.ilo.IloHardware* property), 3129
- supported_network_interfaces
(*ironic.drivers.ipmi.IPMIHardware* property), 3132
- supported_network_interfaces
(*ironic.drivers.irmc.IRMCHardware* property), 3134
- supported_network_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3136
- supported_network_interfaces
(*ironic.drivers.snmp.SNMPHardware* property), 3137
- supported_network_interfaces
(*ironic.drivers.xclarity.XClarityHardware* property), 3153
- supported_power_interfaces
(*ironic.drivers.drac.IDRACHardware* property), 3111
- supported_power_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3115
- supported_power_interfaces
(*ironic.drivers.generic.ManualManagementHardware* property), 3119
- supported_power_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3123
- supported_power_interfaces
(*ironic.drivers.ilo.IloHardware* property), 3129
- supported_power_interfaces
(*ironic.drivers.ipmi.IPMIHardware* property), 3132
- supported_power_interfaces
(*ironic.drivers.irmc.IRMCHardware* property), 3134
- supported_power_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3136
- supported_power_interfaces
(*ironic.drivers.snmp.SNMPHardware* property), 3137
- supported_power_interfaces
(*ironic.drivers.xclarity.XClarityHardware* property), 3153
- supported_raid_interfaces
(*ironic.drivers.drac.IDRACHardware* property), 3111
- supported_raid_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3115

supported_raid_interfaces
(*ironic.drivers.generic.GenericHardware* property), 3117

supported_raid_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3124

supported_raid_interfaces
(*ironic.drivers.ibm.ibmchardware* property), 3126

supported_raid_interfaces
(*ironic.drivers.ilo.ilo5hardware* property), 3127

supported_raid_interfaces
(*ironic.drivers.irmc.irmchardware* property), 3135

supported_raid_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3136

supported_rescue_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3115

supported_rescue_interfaces
(*ironic.drivers.generic.GenericHardware* property), 3118

supported_rescue_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3124

supported_storage_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3115

supported_storage_interfaces
(*ironic.drivers.generic.GenericHardware* property), 3118

supported_storage_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3124

supported_vendor_interfaces
(*ironic.drivers.drac.IDRACHardware* property), 3111

supported_vendor_interfaces
(*ironic.drivers.fake_hardware.FakeHardware* property), 3115

supported_vendor_interfaces
(*ironic.drivers.generic.ManualManagementHardware* property), 3119

supported_vendor_interfaces
(*ironic.drivers.hardware_type.AbstractHardwareType* property), 3124

supported_vendor_interfaces
(*ironic.drivers.ibm.ibmchardware* property), 3126

supported_vendor_interfaces
(*ironic.drivers.ilo.ilo5hardware* property), 3127

supported_vendor_interfaces
(*ironic.drivers.ipmi.IPMIHardware* property), 3132

supported_vendor_interfaces
(*ironic.drivers.redfish.RedfishHardware* property), 3136

supports_is_smartnic()
(*ironic.objects.port.Port* class method), 3336

supports_physical_network()
(*ironic.objects.port.Port* class method), 3336

supports_power_sync()
(*ironic.drivers.base.PowerInterface* method), 3070

supports_power_sync()
(*ironic.drivers.modules.agent_power.AgentPower* method), 2631

supports_version()
(*ironic.objects.base.IronicObject* class method), 3177

swap_mib (*ironic.objects.deployment.Deployment* property), 3241

swift_temp_url()
(*ironic.common.glance_service.image_service.GlanceImage* method), 1087

SwiftAPI (class in *ironic.common.swift*), 1334

SwiftObjectNotFoundError, 1173

SwiftOperationError, 1173

switch_pxe_config() (in module *ironic.drivers.modules.deploy_utils*), 2684

switch_to_tenant_network()
(*ironic.drivers.modules.agent_base.AgentOobStepsMixin* method), 2575

sync() (*ironic.drivers.modules.agent_client.AgentClient* method), 2623

sync_boot_mode() (in module *ironic.drivers.modules.boot_mode_utils*), 2639

sync_node_setting()
(*ironic.objects.bios.BIOSSettingList* class method), 3196

sysobj_oids (*ironic.drivers.modules.snmp.SNMPDriverAuto* attribute), 2932

SYSTEM (in module *ironic.common.components*), 1129

system_id (*ironic.drivers.modules.snmp.SNMPDriverAPCMaster* attribute), 2932

attribute), 2929
 system_id(*ironic.drivers.modules.snmp.SNMPDriverApcPowerSwitch*
 attribute), 2930
 system_id(*ironic.drivers.modules.snmp.SNMPDriverAPCRackPDUs*), 3311
 attribute), 2931
 system_id(*ironic.drivers.modules.snmp.SNMPDriverAte* (*ironic.db.sqlalchemy.models.Node*
 attribute), 2932
 system_id(*ironic.drivers.modules.snmp.SNMPDriverCgcr* target_provision_state
 attribute), 2937
 system_id(*ironic.drivers.modules.snmp.SNMPDriverEatonP386* (*ironic.objects.node.Node* property),
 attribute), 2939
 system_id(*ironic.drivers.modules.snmp.SNMPDriverTeltron* target_provision_state
 attribute), 2941
 property), 3305
 (*ironic.objects.node.NodeSetProvisionStatePayload*
 property), 3311
 (*ironic.db.sqlalchemy.models.Node*
 attribute), 1775
 (*ironic.objects.node.Node* property),
 property), 3286
 target_provision_state
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 property), 3294
 target_provision_state
 (*ironic.objects.node.NodeCRUDPayload*
 property), 3287
 target_provision_state
 (*ironic.objects.node.NodePayload* prop-
 erty), 3300
 target_provision_state
 (*ironic.objects.node.NodeSetPowerStatePayload*
 property), 3305
 target_provision_state
 (*ironic.objects.node.NodeSetProvisionStatePayload*
 property), 3311
 target_raid_config
 (*ironic.db.sqlalchemy.models.Node*
 attribute), 1775
 target_raid_config (*ironic.objects.node.Node*
 property), 3280
 target_state (*ironic.common.fsm.FSM* prop-
 erty), 1183
 TaskManager (class in
ironic.conductor.task_manager), 1563
 tear_down() (*ironic.drivers.base.DeployInterface*
 method), 2997
 tear_down() (*ironic.drivers.modules.agent_base.AgentBaseMixin*
 method), 2557
 tear_down() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
 method), 1941
 tear_down() (*ironic.drivers.modules.fake.FakeDeploy*
 method), 2725
 tear_down_agent()
 (*ironic.drivers.modules.agent_base.AgentDeployMixin*
 method), 2574
 tear_down_agent()
 (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
 method), 1941
 tear_down_cleaning()
 (*ironic.drivers.base.DeployInterface*
 method), 2998

tear_down_cleaning() (in module *ironic.drivers.modules.agent_base.AgentBaseMixin*), 2560
 tear_down_cleaning() (in module *ironic.drivers.modules.ansible.deploy.AnsibleDeployer*), 1942
 tear_down_inband_cleaning() (in module *ironic.drivers.modules.deploy_utils*), 2687
 tear_down_storage_configuration() (in module *ironic.drivers.modules.deploy_utils*), 2689
 tempdir() (in module *ironic.common.utils*), 1364
 template_sanitize() (in module *ironic.api.controllers.v1.deploy_template*), 832
 TemporaryFailure, 1173
 TempUrlCacheElement (class in *ironic.common.glance_service.image_service*), 1090
 TFTPImageCache (class in *ironic.common.pxe_utils*), 1269
 third_method_sync() (in module *ironic.drivers.modules.fake.FakeVendorB*), 2785
 to_event_type_field() (in module *ironic.objects.notification.EventType*), 3312
 to_policy_values() (in module *ironic.common.context.RequestContext*), 1131
 to_power (in module *ironic.objects.node.NodeSetPowerStatePayload*), 3305
 touch() (in module *ironic.objects.conductor.Conductor*), 3216
 touch_conductor() (in module *ironic.db.api.Connection*), 1888
 touch_conductor() (in module *ironic.db.sqlalchemy.api.Connection*), 1735
 touch_node_provisioning() (in module *ironic.db.api.Connection*), 1889
 touch_node_provisioning() (in module *ironic.db.sqlalchemy.api.Connection*), 1735
 touch_provisioning() (in module *ironic.objects.node.Node*), 3280
 trace_cls() (in module *ironic.common.profiler*), 1267
 BaseMixin (class in *ironic.objects.trait*), 3357
 trait (in module *ironic.db.sqlalchemy.models.NodeTrait*), 1776
 TraitList (class in *ironic.objects.trait*), 3362
 traits (in module *ironic.db.sqlalchemy.models.Allocation*), 1766
 traits (in module *ironic.objects.allocation.Allocation*), 3165
 traits (in module *ironic.objects.allocation.AllocationCRUDPayload*), 3168
 traits (in module *ironic.objects.node.Node*), 3280
 traits (in module *ironic.objects.node.NodeCorrectedPowerStatePayload*), 3294
 traits (in module *ironic.objects.node.NodeCRUDPayload*), 3287
 traits (in module *ironic.objects.node.NodePayload*), 3300
 traits (in module *ironic.objects.node.NodeSetPowerStatePayload*), 3305
 traits (in module *ironic.objects.node.NodeSetProvisionStatePayload*), 3311
 translate_from_glance() (in module *ironic.common.glance_service.service_utils*), 1093
 translate_xclarity_power_action() (in module *ironic.drivers.modules.xclarity.common*), 2511
 translate_xclarity_power_state() (in module *ironic.drivers.modules.xclarity.common*), 2511
 try_set_boot_device() (in module *ironic.drivers.modules.deploy_utils*), 2690
 type (in module *ironic.db.sqlalchemy.models.VolumeConnector*), 1779
 type (in module *ironic.objects.volume_connector.VolumeConnector*), 3383
 type (in module *ironic.objects.volume_connector.VolumeConnectorCRUDPayload*), 3385
 types() (in module *ironic.common.args*), 1106
U
 UEFI (in module *ironic.common.boot_modes*), 1114
 umount() (in module *ironic.common.utils*), 1365
 Unauthorized, 1173
 unbind_neutron_port() (in module

- `ironic.common.neutron`), 1250
- `unconfigure_tenant_networks()` (`ironic.drivers.base.NetworkInterface` method), 3057
- `unconfigure_tenant_networks()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 2356
- `unconfigure_tenant_networks()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 2366
- `unconfigure_tenant_networks()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 2374
- UNDEPLOY (in module `ironic.common.states`), 1331
- `unique` (`ironic.db.sqlalchemy.models.BIOSSetting` attribute), 1767
- `unique` (`ironic.objects.bios.BIOSSetting` property), 3189
- `unit_id` (`ironic.drivers.modules.snmp.SNMPDriverBaytechMRP27c` attribute), 2936
- `unix_file_modification_datetime()` (in module `ironic.common.utils`), 1366
- UNKNOWN (in module `ironic.common.indicator_states`), 1216
- UnknownArgument, 1173
- UnknownAttribute, 1173
- `unpublish_image()` (`ironic.drivers.modules.image_utils.ImageHandler` method), 2794
- `unpublish_image_for_node()` (`ironic.drivers.modules.image_utils.ImageHandler` class method), 2795
- `unregister()` (`ironic.objects.conductor.Conductor` method), 3216
- `unregister_all_hardware_interfaces()` (`ironic.objects.conductor.Conductor` method), 3217
- `unregister_conductor()` (`ironic.db.api.Connection` method), 1890
- `unregister_conductor()` (`ironic.db.sqlalchemy.api.Connection` method), 1736
- `unregister_conductor_hardware_interfaces()` (`ironic.db.api.Connection` method), 1890
- `unregister_conductor_hardware_interfaces()` (`ironic.db.sqlalchemy.api.Connection` method), 1737
- `unrescue()` (`ironic.drivers.base.RescueInterface` method), 3086
- `unrescue()` (`ironic.drivers.modules.agent.AgentRescue` method), 2537
- `unrescue()` (`ironic.drivers.modules.fake.FakeRescue` method), 2773
- `unrescue()` (`ironic.drivers.modules.noop.NoRescue` method), 2883
- UNRESCUEFAIL (in module `ironic.common.states`), 1331
- UNRESCUING (in module `ironic.common.states`), 1331
- `unset_node_tags()` (`ironic.db.api.Connection` method), 1891
- `unset_node_tags()` (`ironic.db.sqlalchemy.api.Connection` method), 1738
- `unset_node_traits()` (`ironic.db.api.Connection` method), 1891
- `unset_node_traits()` (`ironic.db.sqlalchemy.api.Connection` method), 1738
- UNSTABLE_STATES (in module `ironic.common.states`), 1331
- UnsupportedDriverExtension, 1174
- `update_allocation()` (`ironic.db.api.Connection` method), 1892
- `update_allocation()` (`ironic.db.sqlalchemy.api.Connection` method), 1739
- UPDATE_ALLOWED_STATES (in module `ironic.common.states`), 1332
- `update_auth_failure_logging_threshold()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 2234
- `update_bios_setting_list()` (`ironic.db.api.Connection` method), 1894
- `update_bios_setting_list()` (`ironic.db.sqlalchemy.api.Connection` method), 1741
- `update_boot_mode()` (in module `ironic.drivers.modules.ilo.common`), 2187
- `update_chassis()` (`ironic.db.api.Connection` method), 1897
- `update_chassis()` (`ironic.db.sqlalchemy.api.Connection` method), 1743
- `update_cipher_suite_cmd()` (in module `ironic.drivers.modules.ipmitool`), 2860
- `update_deploy_template()`

(*ironic.db.api.Connection* method), 1341
 1898

`update_deploy_template()`
 (*ironic.db.sqlalchemy.api.Connection* method), 1744

`update_dhcp()` (*ironic.common.dhcp_factory.DHCPFactory* method), 1133

`update_dhcp_opts()`
 (*ironic.dhcp.base.BaseDHCP* method), 1913

`update_dhcp_opts()`
 (*ironic.dhcp.neutron.NeutronDHCPApi* method), 1920

`update_dhcp_opts()`
 (*ironic.dhcp.none.NoneDHCPApi* method), 1927

`update_driver_config()`
 (*ironic.drivers.modules.image_utils.ImageHandler* method), 2796

`update_firmware()`
 (*ironic.drivers.modules.ilo.management.IloManagement* method), 2235

`update_firmware()`
 (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 2446

`update_firmware_sum()`
 (*ironic.drivers.modules.ilo.management.IloManagement* method), 2236

`update_ipmi_properties()` (in module *ironic.drivers.modules.ilo.common*), 2188

`update_ipmi_properties()` (in module *ironic.drivers.modules.irmc.common*), 2291

`update_minimum_password_length()`
 (*ironic.drivers.modules.ilo.management.IloManagement* method), 2237

`update_neutron_port()` (in module *ironic.common.neutron*), 1251

`update_next_step_index()` (in module *ironic.conductor.utils*), 1613

`update_node()` (*ironic.conductor.manager.ConductorManager* method), 1408

`update_node()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1524

`update_node()` (*ironic.db.api.Connection* method), 1900

`update_node()` (*ironic.db.sqlalchemy.api.Connection* method), 1746

`update_object_meta()`
 (*ironic.common.swift.SwiftAPI* method), 1341

`update_opt_defaults()` (in module *ironic.conf.opts*), 1627

`update_port()` (*ironic.conductor.manager.ConductorManager* method), 1408

`update_port()` (*ironic.conductor.rpcapi.ConductorAPI* method), 1527

`update_port()` (*ironic.db.api.Connection* method), 1902

`update_port()` (*ironic.db.sqlalchemy.api.Connection* method), 1748

`update_port_address()` (in module *ironic.common.neutron*), 1252

`update_port_dhcp_opts()`
 (*ironic.dhcp.base.BaseDHCP* method), 1916

`update_port_dhcp_opts()`
 (*ironic.dhcp.neutron.NeutronDHCPApi* method), 1923

`update_port_dhcp_opts()`
 (*ironic.dhcp.none.NoneDHCPApi* method), 1930

`update_portgroup()`
 (*ironic.conductor.manager.ConductorManager* method), 1408

`update_portgroup()`
 (*ironic.conductor.rpcapi.ConductorAPI* method), 1528

`update_portgroup()` (*ironic.db.api.Connection* method), 1902

`update_portgroup()`
 (*ironic.db.sqlalchemy.api.Connection* method), 1749

`update_raid_info()` (in module *ironic.common.raid*), 1304

`update_state_in_older_versions()` (in module *ironic.api.controllers.v1.node*), 899

`update_to_latest_versions()`
 (*ironic.db.api.Connection* method), 1904

`update_to_latest_versions()`
 (*ironic.db.sqlalchemy.api.Connection* method), 1750

`update_volume_connector()`
 (*ironic.conductor.manager.ConductorManager* method), 1408

`update_volume_connector()`
 (*ironic.conductor.rpcapi.ConductorAPI* method), 1530

`update_volume_connector()`

property), 3317

updated_at (*ironic.objects.port.Port* property), 3336

updated_at (*ironic.objects.port.PortCRUDNotification* property), 3338

updated_at (*ironic.objects.port.PortCRUDPayload* property), 3339

updated_at (*ironic.objects.portgroup.Portgroup* property), 3354

updated_at (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 3356

updated_at (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 3357

updated_at (*ironic.objects.trait.Trait* property), 3362

updated_at (*ironic.objects.trait.TraitList* property), 3367

updated_at (*ironic.objects.volume_connector.VolumeConnector* property), 3383

updated_at (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* property), 3384

updated_at (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 3385

updated_at (*ironic.objects.volume_target.VolumeTarget* property), 3404

updated_at (*ironic.objects.volume_target.VolumeTargetCRUDNotification* property), 3405

updated_at (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406

upgrade

- ironic-dbsync command line option, 537
- command line option
 - help, 540
 - revision <REVISION>, 540
 - h, 540
- upgrade() (*in module ironic.db.migration*), 1911
- upgrade() (*in module ironic.db.sqlalchemy.migration*), 1764
- upgrade() (*ironic.cmd.dbsync.DBCommand* method), 1083
- upgrade_lock() (*ironic.conductor.task_manager.TaskManager* method), 1570
- upper_bound (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 1767
- upper_bound (*ironic.objects.bios.BIOSSetting* property), 3189
- url (*ironic.common.glance_service.image_service.TempUrlCache* attribute), 1090
- url_expires_at

(*ironic.common.glance_service.image_service.TempUrlCache* attribute), 1090

uuid (*ironic.db.sqlalchemy.models.Allocation* attribute), 1766

uuid (*ironic.db.sqlalchemy.models.Chassis* attribute), 1768

uuid (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 1770

uuid (*ironic.db.sqlalchemy.models.Node* attribute), 1775

uuid (*ironic.db.sqlalchemy.models.Port* attribute), 1777

uuid (*ironic.db.sqlalchemy.models.Portgroup* attribute), 1778

uuid (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 1779

uuid (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780

uuid (*ironic.objects.allocation.Allocation* property), 1784

uuid (*ironic.objects.allocation.AllocationCRUDPayload* property), 1784

uuid (*ironic.objects.chassis.Chassis* property), 3206

uuid (*ironic.objects.chassis.ChassisCRUDPayload* property), 3208

uuid (*ironic.objects.deploy_template.DeployTemplate* property), 3231

uuid (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* property), 3233

uuid (*ironic.objects.deployment.Deployment* property), 3242

uuid (*ironic.objects.node.Node* property), 3280

uuid (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 3294

uuid (*ironic.objects.node.NodeCRUDPayload* property), 3287

uuid (*ironic.objects.node.NodePayload* property), 3300

uuid (*ironic.objects.node.NodeSetPowerStatePayload* property), 3305

uuid (*ironic.objects.node.NodeSetProvisionStatePayload* property), 3311

uuid (*ironic.objects.port.Port* property), 3337

uuid (*ironic.objects.port.PortCRUDPayload* property), 3339

uuid (*ironic.objects.portgroup.Portgroup* property), 3354

uuid (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 3357

uuid (*ironic.objects.volume_connector.VolumeConnector*

- property), 3383
- uuid (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 3385
- uuid (*ironic.objects.volume_target.VolumeTarget* property), 3404
- uuid (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406
- uuid() (in module *ironic.common.args*), 1107
- uuid_or_name() (in module *ironic.common.args*), 1108
- UUIDField (class in *ironic.objects.fields*), 3250
- ## V
- validate() (in module *ironic.common.args*), 1110
- validate() (*ironic.api.controllers.v1.node.NodesController* method), 895
- validate() (*ironic.drivers.base.BaseInterface* method), 2974
- validate() (*ironic.drivers.base.NetworkInterface* method), 3057
- validate() (*ironic.drivers.base.RAIDInterface* method), 3080
- validate() (*ironic.drivers.base.VendorInterface* method), 3093
- validate() (*ironic.drivers.modules.agent.AgentDeploy* method), 2524
- validate() (*ironic.drivers.modules.agent.AgentRescue* method), 2539
- validate() (*ironic.drivers.modules.agent.CustomAgentDeploy* method), 2548
- validate() (*ironic.drivers.modules.agent_power.AgentPower* method), 2632
- validate() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 1943
- validate() (*ironic.drivers.modules.drac.bios.DracWSManBIOS* method), 1950
- validate() (*ironic.drivers.modules.drac.inspect.DracWSManInspect* method), 1980
- validate() (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 2004
- validate() (*ironic.drivers.modules.drac.power.DracWSManPower* method), 2014
- validate() (*ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthru* method), 2061
- validate() (*ironic.drivers.modules.fake.FakeBIOS* method), 2703
- validate() (*ironic.drivers.modules.fake.FakeBoot* method), 2710
- validate() (*ironic.drivers.modules.fake.FakeConsole* method), 2716
- validate() (*ironic.drivers.modules.fake.FakeDeploy* method), 2726
- validate() (*ironic.drivers.modules.fake.FakeInspect* method), 2731
- validate() (*ironic.drivers.modules.fake.FakeManagement* method), 2755
- validate() (*ironic.drivers.modules.fake.FakePower* method), 2763
- validate() (*ironic.drivers.modules.fake.FakeRescue* method), 2774
- validate() (*ironic.drivers.modules.fake.FakeStorage* method), 2779
- validate() (*ironic.drivers.modules.fake.FakeVendorA* method), 2782
- validate() (*ironic.drivers.modules.fake.FakeVendorB* method), 2785
- validate() (*ironic.drivers.modules.ibm.management.IBMManagement* method), 2078
- validate() (*ironic.drivers.modules.ibm.power.IBMCPower* method), 2087
- validate() (*ironic.drivers.modules.ibm.vendor.IBMVendor* method), 2104
- validate() (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 2110
- validate() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* method), 2124
- validate() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 2137
- validate() (*ironic.drivers.modules.ilo.console.IloConsoleInterface* method), 2193
- validate() (*ironic.drivers.modules.ilo.inspect.IloInspect* method), 2202
- validate() (*ironic.drivers.modules.ilo.management.IloManagement* method), 2238
- validate() (*ironic.drivers.modules.ilo.power.IloPower* method), 2246
- validate() (*ironic.drivers.modules.ilo.vendor.VendorPassthru* method), 2257
- validate() (*ironic.drivers.modules.inspector.Inspector* method), 2015
- validate() (*ironic.drivers.modules.ipmitool.IPMIConsole* method), 2818
- validate() (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 2826
- validate() (*ironic.drivers.modules.ipmitool.IPMIPower* method), 2839
- validate() (*ironic.drivers.modules.ipmitool.VendorPassthru* method), 2852
- validate() (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 2265
- validate() (*ironic.drivers.modules.irmc.boot.IRMVirtualMedia* method), 2265

method), 2276
 validate() (ironic.drivers.modules.irmc.inspect.IRMCIInspect), 1367
 method), 2293
 validate() (ironic.drivers.modules.irmc.management.IRMCMANAGEMENT), 1298
 method), 2313
 validate() (ironic.drivers.modules.irmc.power.IRMCPower), 2323
 method), 2323
 validate() (ironic.drivers.modules.network.flat.FlatNetwork), 2692
 method), 2357
 validate() (ironic.drivers.modules.network.neutron.NeutronNetwork), 1369
 method), 2367
 validate() (ironic.drivers.modules.noop.FailMixin), 1305
 method), 2863
 validate() (ironic.drivers.modules.noop_mgmt.NoopManagement), 1392
 method), 2899
 validate() (ironic.drivers.modules.pxe.PXEAnacondaDeploy), 1408
 method), 2906
 validate() (ironic.drivers.modules.pxe.PXERamdiskDeploy), 1537
 method), 2912
 validate() (ironic.drivers.modules.pxe_base.PXEBaseMixin), 1537
 method), 2920
 validate() (ironic.drivers.modules.redfish.bios.RedfishBIOS), 1186
 method), 2386
 validate() (ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot), 1189
 method), 2399
 validate() (ironic.drivers.modules.redfish.inspect.RedfishInspect), 1189
 method), 2405
 validate() (ironic.drivers.modules.redfish.management.RedfishManagement), 1193
 method), 2447
 validate() (ironic.drivers.modules.redfish.power.RedfishPower), 2551
 method), 2456
 validate() (ironic.drivers.modules.redfish.raid.RedfishRAID), 2551
 method), 2470
 validate() (ironic.drivers.modules.redfish.vendor.RedfishVendorPassives), 2693
 method), 2488
 validate() (ironic.drivers.modules.snmp.SNMPPower), 2552
 method), 2949
 validate() (ironic.drivers.modules.storage.cinder.CinderStorage), 1233
 method), 2495
 validate() (ironic.drivers.modules.storage.external.ExternalStorage), 1233
 method), 2499
 validate() (ironic.drivers.modules.storage.noop.NoopStorage), 2981
 method), 2505
 validate() (ironic.drivers.modules.xclarity.manager.XClarityManager), 3058
 method), 2517
 validate() (ironic.drivers.modules.xclarity.power.XClarityPower), 3058
 method), 2522
 validate() (ironic.drivers.utils.MixinVendorInterface), 2126
 method), 3138
 validate_and_normalize_datapath_id() (in module ironic.common.utils), 1366
 validate_and_normalize_mac() (in module ironic.common.utils), 1367
 validate_boot_parameters_for_trusted_boot() (in module ironic.common.pxe_utils), 1298
 validate_capabilities() (in module ironic.drivers.modules.deploy_utils), 2692
 validate_conductor_group() (in module ironic.common.utils), 1369
 validate_configuration() (in module ironic.common.raid), 1305
 validate_deploy_steps() (in module ironic.conductor.manager.ConductorManager), 1408
 validate_driver_interfaces() (in module ironic.common.image_service.BaseImageService), 1186
 validate_driver_interfaces() (in module ironic.common.image_service.FileImageService), 1189
 validate_driver_interfaces() (in module ironic.common.image_service.HttpImageService), 1193
 validate_href() (in module ironic.common.image_service.BaseImageService), 1186
 validate_href() (in module ironic.common.image_service.FileImageService), 1189
 validate_href() (in module ironic.common.image_service.HttpImageService), 1193
 validate_http_provisioning_configuration() (in module ironic.drivers.modules.agent), 2551
 validate_image_properties() (in module ironic.drivers.modules.deploy_utils), 2693
 validate_image_proxies() (in module ironic.drivers.modules.agent), 2552
 validate_inspection() (in module ironic.common.neutron.NeutronNetworkInterfaceMixin), 1233
 validate_inspection() (in module ironic.drivers.base.BootInterface), 2981
 validate_inspection() (in module ironic.drivers.base.NetworkInterface), 3058
 validate_inspection() (in module ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot), 2126
 validate_inspection() (in module ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot), 2126

- method*), 2138
- `validate_inspection()`
(*ironic.drivers.modules.network.noop.NoopNetworkInterface*
method), 2375
- `validate_inspection()`
(*ironic.drivers.modules.pxe_base.PXEBaseMixin*
method), 2922
- `validate_inspection()`
(*ironic.drivers.modules.redfish.boot.RedfishBootInterface*
method), 2400
- `validate_instance_info_traits()` (*in module*
ironic.conductor.utils), 1614
- `validate_job_queue()` (*in module*
ironic.drivers.modules.drac.job), 1983
- `validate_kickstart_file()` (*in module*
ironic.common.pxe_utils), 1298
- `validate_kickstart_template()` (*in module*
ironic.common.pxe_utils), 1299
- `validate_limit()` (*in module*
ironic.api.controllers.v1.utils), 1019
- `validate_network()` (*in module*
ironic.common.neutron), 1253
- `validate_network_data()` (*in module*
ironic.api.controllers.v1.node), 900
- `validate_network_port()` (*in module*
ironic.common.utils), 1369
- `validate_node()` (*in module*
ironic.conductor.deployments), 1392
- `validate_port_info()` (*in module*
ironic.common.neutron), 1255
- `validate_port_physnet()` (*in module*
ironic.conductor.utils), 1615
- `validate_raid_config()`
(*ironic.drivers.base.RAIDInterface*
method), 3081
- `validate_raid_config()`
(*ironic.drivers.modules.noop.NoRAID*
method), 2880
- `validate_rescue()`
(*ironic.drivers.base.BootInterface*
method), 2982
- `validate_rescue()`
(*ironic.drivers.base.NetworkInterface*
method), 3060
- `validate_rescue()`
(*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
method), 2127
- `validate_rescue()`
(*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*
method), 2139
- `validate_rescue()`
(*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot*
method), 2278
- `validate_rescue()`
(*ironic.drivers.modules.network.neutron.NeutronNetworkInterface*
method), 2368
- `validate_rescue()`
(*ironic.drivers.modules.pxe_base.PXEBaseMixin*
method), 2922
- `validate_security_parameter_values()`
(*in module*
ironic.drivers.modules.ilo.common),
2188
- `validate_sort_dir()` (*in module*
ironic.api.controllers.v1.utils), 1019
- `validate_user_deploy_steps_and_templates()`
(*in module* *ironic.conductor.steps*), 1552
- `value` (*ironic.db.sqlalchemy.models.BIOSSetting*
attribute), 1767
- `value` (*ironic.objects.bios.BIOSSetting* *property*),
3189
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSw*
attribute), 2929
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSw*
attribute), 2930
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU*
attribute), 2931
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverAten*
attribute), 2932
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverBaytechMRP27*
attribute), 2936
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverCyberPower*
attribute), 2937
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverEatonPower*
attribute), 2939
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverSimple*
property), 2940
- `value_power_off`
(*ironic.drivers.modules.snmp.SNMPDriverTeltronix*
attribute), 2942
- `value_power_on`
(*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSw*
attribute), 2929
- `value_power_on`

- [\(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitchAPI.controllers.v1.node.NodesController attribute\), 2930](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverAPCRackPDUC.api.controllers.v1.utils\), 1019 attribute\), 2931](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverAten attribute\), 2932](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverBaytechMPP27d\), 1538 attribute\), 2936](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverCyberPower attribute\), 2937](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverEatonPowerware attribute\), 2939](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverSimple attribute\), 2941 property\), 2941](#)
- [value_power_on \(ironic.drivers.modules.snmp.SNMPDriverTerafit attribute\), 2942](#)
- [value_within_timeout\(\) \(in module ironic.conductor.utils\), 1618](#)
- [vendor \(ironic.drivers.base.BareDriver attribute\), 2964](#)
- [vendor_interface \(ironic.db.sqlalchemy.models.Node attribute\), 1775](#)
- [vendor_interface \(ironic.objects.node.Node property\), 3281](#)
- [vendor_interface \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 3294](#)
- [vendor_interface \(ironic.objects.node.NodeCRUDPayload property\), 3287](#)
- [vendor_interface \(ironic.objects.node.NodePayload property\), 3300](#)
- [vendor_interface \(ironic.objects.node.NodeSetPowerStatePayload property\), 3305](#)
- [vendor_interface \(ironic.objects.node.NodeSetProvisionStatePayload property\), 3311](#)
- [vendor_passthru \(ironic.api.controllers.v1.driver.DriversController attribute\), 839](#)
- [vendor_passthru \(ironic.api.controllers.v1.node.NodesController attribute\), 896](#)
- [vendor_passthru\(\) \(in module ironic.conductor.manager.ConductorManager method\), 1408](#)
- [vendor_passthru\(\) \(ironic.conductor.rpcapi.ConductorAPI method\), 1538](#)
- [VendorInterface \(class in ironic.drivers.base\), 3090](#)
- [VendorMetadata \(class in ironic.drivers.base\), 3095](#)
- [VendorPassthru \(class in ironic.drivers.modules.ilo.vendor\), 2254](#)
- [VendorPassthru \(class in ironic.drivers.modules.ipmitool\), 2848](#)
- [VendorPassthruException, 1174](#)
- [VERBS \(in module ironic.common.states\), 1332](#)
- [verify_firmware_update_args\(\) \(in module ironic.drivers.modules.ilo.firmware_processor\), 2199](#)
- [verify_image_checksum\(\) \(in module ironic.drivers.modules.ilo.common\), 2190](#)
- [verify_node_for_deallocation\(\) \(in module ironic.conductor.allocations\), 1376](#)
- [VERIFYING \(in module ironic.common.states\), 1333](#)
- [version ironic-dbsync command line option, 1054](#)
- [Version \(class in ironic.api.controllers.base\), 1054](#)
- [version \(ironic.db.sqlalchemy.models.Allocation attribute\), 1766](#)
- [version \(ironic.db.sqlalchemy.models.BIOSSetting attribute\), 1767](#)
- [version \(ironic.db.sqlalchemy.models.Chassis attribute\), 1768](#)
- [version \(ironic.db.sqlalchemy.models.Conductor attribute\), 1769](#)
- [version \(ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute\), 1770](#)
- [version \(ironic.db.sqlalchemy.models.DeployTemplate attribute\), 1770](#)
- [version \(ironic.db.sqlalchemy.models.DeployTemplateStep attribute\), 1771](#)
- [version \(ironic.db.sqlalchemy.models.IronicBase attribute\), 1772](#)

- attribute), 1772
- version (*ironic.db.sqlalchemy.models.Node* attribute), 1775
- version (*ironic.db.sqlalchemy.models.NodeTag* attribute), 1775
- version (*ironic.db.sqlalchemy.models.NodeTrait* attribute), 1776
- version (*ironic.db.sqlalchemy.models.Port* attribute), 1777
- version (*ironic.db.sqlalchemy.models.Portgroup* attribute), 1778
- version (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 1779
- version (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780
- VERSION (*ironic.objects.allocation.Allocation* attribute), 3154
- VERSION (*ironic.objects.allocation.AllocationCRUDNotification* attribute), 3166
- VERSION (*ironic.objects.allocation.AllocationCRUDPayload* attribute), 3167
- VERSION (*ironic.objects.bios.BIOSSetting* attribute), 3182
- VERSION (*ironic.objects.bios.BIOSSettingList* attribute), 3189
- VERSION (*ironic.objects.chassis.Chassis* attribute), 3197
- VERSION (*ironic.objects.chassis.ChassisCRUDNotification* attribute), 3207
- VERSION (*ironic.objects.chassis.ChassisCRUDPayload* attribute), 3207
- VERSION (*ironic.objects.conductor.Conductor* attribute), 3208
- VERSION (*ironic.objects.deploy_template.DeployTemplate* attribute), 3217
- VERSION (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* attribute), 3231
- VERSION (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 3232
- VERSION (*ironic.objects.deployment.Deployment* attribute), 3233
- VERSION (*ironic.objects.node.Node* attribute), 3262
- VERSION (*ironic.objects.node.NodeConsoleNotification* attribute), 3287
- VERSION (*ironic.objects.node.NodeCorrectedPowerStateNotification* attribute), 3289
- VERSION (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 3291
- VERSION (*ironic.objects.node.NodeCRUDNotification* attribute), 3281
- VERSION (*ironic.objects.node.NodeCRUDPayload* attribute), 3283
- VERSION (*ironic.objects.node.NodeMaintenanceNotification* attribute), 3295
- VERSION (*ironic.objects.node.NodePayload* attribute), 3296
- VERSION (*ironic.objects.node.NodeSetPowerStateNotification* attribute), 3300
- VERSION (*ironic.objects.node.NodeSetPowerStatePayload* attribute), 3301
- VERSION (*ironic.objects.node.NodeSetProvisionStateNotification* attribute), 3306
- VERSION (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 3307
- VERSION (*ironic.objects.notification.EventType* attribute), 3312
- VERSION (*ironic.objects.notification.NotificationBase* attribute), 3314
- VERSION (*ironic.objects.notification.NotificationPayloadBase* attribute), 3315
- VERSION (*ironic.objects.notification.NotificationPublisher* attribute), 3317
- VERSION (*ironic.objects.port.Port* attribute), 3318
- VERSION (*ironic.objects.port.PortCRUDNotification* attribute), 3337
- VERSION (*ironic.objects.port.PortCRUDPayload* attribute), 3338
- VERSION (*ironic.objects.portgroup.Portgroup* attribute), 3340
- VERSION (*ironic.objects.portgroup.PortgroupCRUDNotification* attribute), 3355
- VERSION (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 3356
- VERSION (*ironic.objects.trait.Trait* attribute), 3357
- VERSION (*ironic.objects.trait.TraitList* attribute), 3360
- VERSION (*ironic.objects.volume_connector.VolumeConnector* attribute), 3367
- VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* attribute), 3384
- VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 3385
- VERSION (*ironic.objects.volume_target.VolumeTarget* attribute), 3386
- VERSION (*ironic.objects.volume_target.VolumeTargetCRUDNotification* attribute), 3404
- VERSION (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 3405
- version command line option
- help, 540
- h, 540

version() (in module *ironic.db.migration*), 1911
 version() (in module *ironic.db.sqlalchemy.migration*), 1764
 version() (*ironic.cmd.dbsync.DBCommand* method), 1083
 VersionSelectorApplication (class in *ironic.api.app*), 1064
 vif_attach() (*ironic.conductor.manager.ConductorManager* method), 1408
 vif_attach() (*ironic.conductor.rpcapi.ConductorAPN* method), 1542
 vif_attach() (*ironic.drivers.base.NetworkInterface* method), 3061
 vif_attach() (*ironic.drivers.modules.network.common* method), 2335
 vif_attach() (*ironic.drivers.modules.network.noop* method), 2376
 vif_detach() (*ironic.conductor.manager.ConductorManager* method), 1408
 vif_detach() (*ironic.conductor.rpcapi.ConductorAPN* method), 1544
 vif_detach() (*ironic.drivers.base.NetworkInterface* method), 3062
 vif_detach() (*ironic.drivers.modules.network.common* method), 2338
 vif_detach() (*ironic.drivers.modules.network.noop* method), 2377
 vif_list() (*ironic.conductor.manager.ConductorManager* method), 1408
 vif_list() (*ironic.conductor.rpcapi.ConductorAPN* method), 1546
 vif_list() (*ironic.drivers.base.NetworkInterface* method), 3063
 vif_list() (*ironic.drivers.modules.network.common* method), 2341
 vif_list() (*ironic.drivers.modules.network.noop* method), 2378
 VifAlreadyAttached, 1174
 VifInvalidForAttach, 1175
 VifNotAttached, 1175
 VIFPortIDMixin (class in *ironic.drivers.modules.network.common*), 2339
 volume_connectors (*ironic.conductor.task_manager.TaskManager* property), 1571
 volume_create_error_handler() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 2472
 volume_id (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780
 volume_id (*ironic.objects.volume_target.VolumeTarget* property), 3404
 volume_id (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406
 volume_targets (*ironic.conductor.task_manager.TaskManager* property), 1571
 VolumeType (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 1780
 Volume_type (*ironic.objects.volume_target.VolumeTarget* property), 3404
 volume_type (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 3406
 VolumeConnectorEFBAIDMixin (class in *ironic.db.sqlalchemy.models*), 1778
 VolumeConnector (class in *ironic.objects.volume_connector*), 3367
 VolumeConnectorAlreadyExists, 1175
 VolumeConnectorCRUDNotification (class in *ironic.objects.volume_connector*), 3383
 VolumeConnectorCRUDPayload (class in *ironic.objects.volume_connector*), 3384
 VolumeConnectorEFBAIDMixin, 1175
 VolumeConnectorsController (class in *ironic.api.controllers.v1.volume_connector*), 1023
 VolumeConnectorTypeAndIdAlreadyExists, 1175
 VolumeController (class in *ironic.api.controllers.v1.volume*), 1023
 VolumeTarget (class in *ironic.db.sqlalchemy.models*), 1779
 VolumeTargetMixin (class in *ironic.objects.volume_target*), 3386
 VolumeTargetAlreadyExists, 1175
 VolumeTargetBootIndexAlreadyExists, 1175
 VolumeTargetCRUDNotification (class in *ironic.objects.volume_target*), 3404
 VolumeTargetCRUDPayload (class in *ironic.objects.volume_target*), 3405
 VolumeTargetNotFound, 1176
 VolumeTargetsController (class in *ironic.api.controllers.v1.volume_target*), 1038

W

wait() (*ironic.common.wsgi_service.WSGIService* method), 1371
 wait_for_host_agent() (in module

ironic.common.neutron), 1256
 wait_for_job_completion() (in module *ironic.drivers.modules.drac.job*), 1984
 wait_for_port_status() (in module *ironic.common.neutron*), 1258
 wait_until_get_system_ready() (in module *ironic.drivers.modules.redfish.utils*), 2485
 WANBOOT (in module *ironic.common.boot_devices*), 1112
 warn_about_unsafe_shred_parameters() (in module *ironic.cmd.conductor*), 1081
 WARNING (*ironic.objects.fields.NotificationLevel* attribute), 3247
 wipe_cleaning_internal_info() (in module *ironic.conductor.utils*), 1619
 wipe_deploy_internal_info() (in module *ironic.conductor.utils*), 1619
 wipe_internal_info_on_power_off() (in module *ironic.conductor.utils*), 1620
 wipe_token_and_url() (in module *ironic.conductor.utils*), 1620
 wrap_ipv6() (in module *ironic.common.utils*), 1370
 wrapfunc() (in module *ironic.api.functions*), 1071
 write_image() (*ironic.drivers.modules.agent.AgentDeploy* method), 2525
 write_image() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 1943
 write_to_file() (in module *ironic.common.utils*), 1370
 WSGIService (class in *ironic.common.wsgi_service*), 1370

X

XClarityError, 1176
 XClarityHardware (class in *ironic.drivers.xclarity*), 3153
 XClarityManagement (class in *ironic.drivers.modules.xclarity.management*), 2512
 XClarityPower (class in *ironic.drivers.modules.xclarity.power*), 2518