
Cyborg Documentation

Release 5.0.1.dev12

OpenStack Foundation

Aug 19, 2021

CONTENTS

1	Overview	2
1.1	Introduction	2
1.1.1	Background Story	2
1.1.2	Definition Breakdown	2
1.2	Cyborg architecture	3
1.3	Usage	3
2	Documentation for Operators	4
2.1	Installation	4
2.1.1	Installation with pip	4
2.1.2	Install Cyborg from Source	7
2.1.3	Installing Cyborg API via WSGI	10
2.2	Configuration Reference	11
2.2.1	Configuration Guide	11
2.2.2	Cyborg Support Matrix	78
3	For End Users	80
3.1	Tools for using Cyborg	80
3.1.1	Command-Line Interface Reference	80
3.2	Using the API	81
4	Documentation for Developers	82
4.1	So You Want to Contribute	82
4.1.1	Communication	82
4.1.2	Contacting the Core Team	82
4.1.3	New Feature Planning	82
4.1.4	Task Tracking	82
4.1.5	Reporting a Bug	83
4.1.6	Getting Your Patch Merged	83
4.1.7	Project Team Lead Duties	83
4.2	DevStack Quick Start	83
4.2.1	Create stack user (optional)	83
4.2.2	Download DevStack	83
4.2.3	Create local.conf file	84
4.2.4	Multi-Node Lab	85
4.2.5	Run DevStack	86
4.2.6	Use OpenStack	86
4.3	Driver Development Guide	86
4.3.1	Derive a new driver class	86

4.3.2	Modify setup.cfg	87
4.3.3	Reinstall and Test	87
5	Indices and tables	88

Cyborg is a general management framework for accelerators

OVERVIEW

1.1 Introduction

1.1.1 Background Story

OpenStack Acceleration Discussion Started from Telco Requirements:

- High level requirements first drafted in the standard organization ETSI NFV ISG
- High level requirements transformed into detailed requirements in OPNFV DPACC project.
- New project called Nomad established to address the requirements.
- BoF discussions back in OpenStack Austin Summit.

Transition to Cyborg Project:

- From a long period of conversation and discussion within the OpenStack community, we found that the initial goal of Nomad project to address acceleration management in Telco is too limited. From design summit session in Barcelona Summit, we have developers from Scientific WG help us understanding the need for acceleration management in HPC cloud, and we also had a lot of discussion on the Public Cloud support of accelerated instances.
- We decide to formally establish a project that will work on the management framework for dedicated devices in OpenStack, and there comes the Cyborg Project.

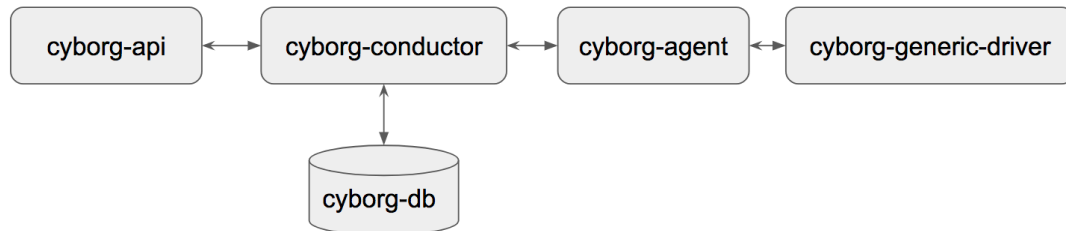
1.1.2 Definition Breakdown

General Management Framework: * Resource Discovery * Life Cycle Management

Accelerators: * Software: dpdk/spdk, pmem, * Hardware: FPGA, GPU, ARM SoC, NVMe SSD, CCIX based Caches,

1.2 Cyborg architecture

Cyborg design can be described by following diagram:



cyborg-api - cyborg-api is a cyborg service that provides **REST API** interface for the Cyborg project. It supports POST/PUT/DELETE/GET operations and interacts with cyborg-agent and cyborg-db via cyborg-conductor.

cyborg-conductor - cyborg-conductor is a cyborg service that coordinates interaction, DB access between cyborg-api and cyborg-agent.

cyborg-agent - cyborg-agent is a cyborg service that is responsible for interaction with accelerator backends via the Cyborg Driver. For now the only implementation in play is the Cyborg generic Driver. It will also handle the communication with the Nova placement service. Cyborg-Agent will also write to a local cache for local accelerator events.

cyborg-generic-driver - cyborg-generic-driver is a general multipurpose driver with the common set of capabilities that any accelerators will have.

1.3 Usage

To use cyborg in a project:

```
import cyborg
```

DOCUMENTATION FOR OPERATORS

The documentation in this section is aimed at Cloud Operators needing to install or configure Cyborg.

2.1 Installation

2.1.1 Installation with pip

At the command line:

```
$ pip install openstack-cyborg
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv cyborg
$ pip install openstack-cyborg
```

Common Configuration

Regardless of the package or code source you must do the following to properly setup the Accelerator Life Cycle Management service. A database, service credentials, and API endpoints must be created.

1. To create the database, complete these steps:

- Use the database access client to connect to the database server as the `root` user:

```
$ mysql -u root -p
```

- Create the `cyborg` database:

```
CREATE DATABASE cyborg;
```

- Grant proper access to the `cyborg` database:

```
GRANT ALL PRIVILEGES ON cyborg.* TO 'cyborg'@'localhost'
↳ IDENTIFIED BY 'CYBORG_DBPASS';
```

Replace `CYBORG_DBPASS` with a suitable password.

- Exit the database access client.

```
exit;
```

2. Source the admin credentials to gain access to admin-only CLI commands:

```
$ . admin-openrc
```

3. To create the service credentials, complete these steps:

- Create the cyborg user:

```
$ openstack user create --domain default --password-prompt cyborg
```

- Add the admin role to the cyborg user:

```
$ openstack role add --project service --user cyborg admin
```

- Create the cyborg service entities:

```
$ openstack service create --name cyborg --description  
↪ "Acceleration Service" accelerator
```

4. Create the Acceleration service API endpoints:

- If cyborg-api service is deployed using uwsgi, create the following endpoints:

```
$ openstack endpoint create --region RegionOne \  
accelerator public http://<cyborg-ip>/accelerator/v1  
$ openstack endpoint create --region RegionOne \  
accelerator internal http://<cyborg-ip>/accelerator/v1  
$ openstack endpoint create --region RegionOne \  
accelerator admin http://<cyborg-ip>/accelerator/v1
```

- Otherwise, if cyborg-api service is running as a python process, create the following endpoints:

```
$ openstack endpoint create --region RegionOne \  
accelerator public http://<cyborg-ip>:6666/v1  
$ openstack endpoint create --region RegionOne \  
accelerator internal http://<cyborg-ip>:6666/v1  
$ openstack endpoint create --region RegionOne \  
accelerator admin http://<cyborg-ip>:6666/v1
```

Note: URLs (publicurl, internalurl and adminurl) may be different depending on your environment.

Configure Cyborg

1. Edit `cyborg.conf` with your favorite editor. Below is an example which contains basic settings you likely need to configure.

```
[DEFAULT]
transport_url = rabbit://%RABBITMQ_USER%:%RABBITMQ_PASSWORD%@
↳%OPENSTACK_HOST_IP%:5672/
use_syslog = False
state_path = /var/lib/cyborg
debug = True

...

[database]
connection = mysql+pymysql://%DATABASE_USER%:%DATABASE_PASSWORD%@
↳%OPENSTACK_HOST_IP%/cyborg

...

[service_catalog]
cafile = /opt/stack/data/ca-bundle.pem
project_domain_id = default
user_domain_id = default
project_name = service
password = cyborg
username = cyborg
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password

...

[placement]
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = placement
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password
auth_section = keystone_authtoken

...

[keystone_authtoken]
memcached_servers = localhost:11211
signing_dir = /var/cache/cyborg/api
cafile = /opt/stack/data/ca-bundle.pem
project_domain_name = Default
project_name = service
user_domain_name = Default
password = cyborg
username = cyborg
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password
```

2. Create database tables for Cyborg.

```
cyborg-dbsync --config-file /etc/cyborg/cyborg.conf upgrade
```

3. Install Cyborg API via WSGI *api-uwsgi*

Note: Cyborg-api service can also be run as a Python command that runs a web serve, which can be launched as follows with different Acceleration service API endpoints as mentioned in Prerequisites part. However, we would like to recommend you the uwsgi way since when a project provides a WSGI application the API service gains flexibility in terms of deployment, performance, configuration and scaling. BYW, if you choose devstack to deploy your acceleration service, uwsgi is a default choice.

cyborg-api config-file=/etc/cyborg/cyborg.conf

1. Launch Cyborg Conductor, Cyborg Agent services. Open a separate terminal for each service since the console will be locked by a running process.

```
cyborg-conductor --config-file=/etc/cyborg/cyborg.conf  
cyborg-agent --config-file=/etc/cyborg/cyborg.conf
```

2.1.2 Install Cyborg from Source

This section describes how to install and configure the Acceleration Service for Ubuntu 18.04.1 LTS from source code.

Install from git repository

1. Create a folder which will hold all Cyborg components.

```
mkdir ~/cyborg
```

2. Clone the cyborg git repository to the management server.

```
cd ~/cyborg  
git clone https://opendev.org/openstack/cyborg
```

3. Set up the cyborg config file

First, generate a sample configuration file, using tox

```
cd ~/cyborg/cyborg  
tox -e genconfig
```

And make a copy of it for further modifications

```
cp -r ~/cyborg/cyborg/etc/cyborg /etc  
cd /etc/cyborg  
ln -s cyborg.conf.sample cyborg.conf
```

4. Install Cyborg packages.

```
cd ~/cyborg/cyborg  
sudo python setup.py install
```

Common Configuration

Regardless of the package or code source you must do the following to properly setup the Accelerator Life Cycle Management service. A database, service credentials, and API endpoints must be created.

1. To create the database, complete these steps:

- Use the database access client to connect to the database server as the `root` user:

```
$ mysql -u root -p
```

- Create the `cyborg` database:

```
CREATE DATABASE cyborg;
```

- Grant proper access to the `cyborg` database:

```
GRANT ALL PRIVILEGES ON cyborg.* TO 'cyborg'@'localhost'
↳ IDENTIFIED BY 'CYBORG_DBPASS';
```

Replace `CYBORG_DBPASS` with a suitable password.

- Exit the database access client.

```
exit;
```

2. Source the admin credentials to gain access to admin-only CLI commands:

```
$ . admin-openrc
```

3. To create the service credentials, complete these steps:

- Create the `cyborg` user:

```
$ openstack user create --domain default --password-prompt cyborg
```

- Add the admin role to the `cyborg` user:

```
$ openstack role add --project service --user cyborg admin
```

- Create the `cyborg` service entities:

```
$ openstack service create --name cyborg --description
↳ "Acceleration Service" accelerator
```

4. Create the Acceleration service API endpoints:

- If `cyborg-api` service is deployed using `uwsgi`, create the following endpoints:

```
$ openstack endpoint create --region RegionOne \
accelerator public http://<cyborg-ip>/accelerator/v1
$ openstack endpoint create --region RegionOne \
accelerator internal http://<cyborg-ip>/accelerator/v1
$ openstack endpoint create --region RegionOne \
accelerator admin http://<cyborg-ip>/accelerator/v1
```

- Otherwise, if `cyborg-api` service is running as a python process, create the following endpoints:

```
$ openstack endpoint create --region RegionOne \
  accelerator public http://<cyborg-ip>:6666/v1
$ openstack endpoint create --region RegionOne \
  accelerator internal http://<cyborg-ip>:6666/v1
$ openstack endpoint create --region RegionOne \
  accelerator admin http://<cyborg-ip>:6666/v1
```

Note: URLs (publicurl, internalurl and adminurl) may be different depending on your environment.

Configure Cyborg

1. Edit `cyborg.conf` with your favorite editor. Below is an example which contains basic settings you likely need to configure.

```
[DEFAULT]
transport_url = rabbit://%RABBITMQ_USER%:%RABBITMQ_PASSWORD%@
↳%OPENSTACK_HOST_IP%:5672/
use_syslog = False
state_path = /var/lib/cyborg
debug = True

...

[database]
connection = mysql+pymysql://%DATABASE_USER%:%DATABASE_PASSWORD%@
↳%OPENSTACK_HOST_IP%/cyborg

...

[service_catalog]
cafile = /opt/stack/data/ca-bundle.pem
project_domain_id = default
user_domain_id = default
project_name = service
password = cyborg
username = cyborg
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password

...

[placement]
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = placement
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password
auth_section = keystone_auth_token

...
```

(continues on next page)

(continued from previous page)

```
[keystone_authtoken]
memcached_servers = localhost:11211
signing_dir = /var/cache/cyborg/api
cafile = /opt/stack/data/ca-bundle.pem
project_domain_name = Default
project_name = service
user_domain_name = Default
password = cyborg
username = cyborg
auth_url = http://%OPENSTACK_HOST_IP%/identity
auth_type = password
```

2. Create database tables for Cyborg.

```
cyborg-dbsync --config-file /etc/cyborg/cyborg.conf upgrade
```

3. Install Cyborg API via WSGI *api-uwsgi*

Note: Cyborg-api service can also be run as a Python command that runs a web serve, which can be launched as follows with different Acceleration service API endpoints as mentioned in Prerequisites part. However, we would like to recommend you the uwsgi way since when a project provides a WSGI application the API service gains flexibility in terms of deployment, performance, configuration and scaling. BYW, if you choose devstack to deploy your acceleration service, uwsgi is a default choice.

```
cyborg-api config-file=/etc/cyborg/cyborg.conf
```

1. Launch Cyborg Conductor, Cyborg Agent services. Open a separate terminal for each service since the console will be locked by a running process.

```
cyborg-conductor --config-file=/etc/cyborg/cyborg.conf
cyborg-agent --config-file=/etc/cyborg/cyborg.conf
```

2.1.3 Installing Cyborg API via WSGI

Cyborg-api service can be run either as a Python command that runs a web serve or As a WSGI application hosted by uwsgi. This document is a guide to deploy cyborg-api using uwsgi. In devstack, uwsgi is used by default for development.

WSGI Application

The function `cyborg.api.wsgi_app.init_application` will setup a WSGI application to run behind uwsgi.

Watcher API behind uwsgi

Create a `cyborg-api-uwsgi` file with content below:

```
[uwsgi]
chmod-socket = 666
socket = /var/run/uwsgi/cyborg-wsgi-api.socket
lazy-apps = true
add-header = Connection: close
buffer-size = 65535
hook-master-start = unix_signal:15 gracefully_kill_them_all
thunder-lock = true
plugins = python
enable-threads = true
worker-reload-mercy = 90
exit-on-reload = false
die-on-term = true
master = true
processes = 2
wsgi-file = /usr/local/bin/cyborg-wsgi-api
```

Start `cyborg-api`:

```
# uwsgi --ini /etc/cyborg/cyborg-api-uwsgi.ini
```

2.2 Configuration Reference

2.2.1 Configuration Guide

Configuration options for the Acceleration service

The following options can be set in the `/etc/cyborg/cyborg.conf` config file. A *sample configuration file* is also available.

DEFAULT

`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.

`host`

Type host address

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, a valid hostname, FQDN, or IP address.

periodic_interval

Type integer

Default 60

Default interval (in seconds) for running periodic tasks.

thread_pool_size

Type integer

Default 10

This option specifies the size of the pool of threads used by API to do async jobs. It is possible to limit the number of concurrent connections using this option.

bind_timeout

Type integer

Default 60

This option specifies the timeout of async job for ARQ bind.

pybasedir

Type string

Default /usr/lib/python/site-packages/cyborg/cyborg

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Directory where the cyborg python module is installed.

bindir

Type string

Default \$pybasedir/bin

Directory where cyborg binaries are installed.

state_path

Type string

Default \$pybasedir

Top-level directory for maintaining cyborgs state.

rpc_conn_pool_size

Type integer

Default 30

Minimum Value 1

Size of RPC connection pool.

Table 1: 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 2: 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.

`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 3: Deprecated Variations

Group	Name
DEFAULT	<code>log-config</code>
DEFAULT	<code>log_config</code>

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 4: Deprecated Variations

Group	Name
DEFAULT	<code>logfile</code>

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 5: 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=WARN', 'amqplib=WARN', 'boto=WARN',
'qpuid=WARN', 'sqlalchemy=WARN', 'suds=INFO',
'oslo.messaging=INFO', 'oslo_messaging=INFO',
'iso8601=WARN', 'requests.packages.urllib3.
connectionpool=WARN', 'urllib3.connectionpool=WARN',
'websocket=WARN', 'requests.packages.
urllib3.util.retry=WARN', 'urllib3.util.
retry=WARN', 'keystonemiddleware=WARN', 'routes.
middleware=WARN', 'stevedore=WARN', 'taskflow=WARN',
'keystoneauth=WARN', 'oslo.cache=INFO',
'oslo_policy=INFO', 'dogpile.core.dogpile=INFO']`

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.

agent

enabled_drivers

Type list

Default []

The accelerator drivers enabled on this agent. Such as intel_fpga_driver, in-spur_fpga_driver, nvidia_gpu_driver, etc.

api

host_ip

Type host address

Default 127.0.0.1

The IP address on which cyborg-api listens.

port

Type port number

Default 6666

Minimum Value 0

Maximum Value 65535

The TCP port on which cyborg-api listens.

api_workers

Type integer

Default <None>

Number of workers for OpenStack Cyborg API service. The default is equal to the number of CPUs available if that can be determined, else a default worker count of 1 is returned.

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 change public API endpoint to represent SSL termination URL with public_endpoint option.

public_endpoint

Type string

Default <None>

Public URL to use when building the links to the API resources (for example, <https://cyborg.rocks:6666>). 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 proxys URL. Defaults to None.

api_paste_config**Type** string**Default** `api-paste.ini`

Configuration file for WSGI definition of API.

database**mysql_engine****Type** string**Default** InnoDB

MySQL engine to use.

sqlite_synchronous**Type** boolean**Default** True

If True, SQLite uses synchronous mode.

Table 6: Deprecated Variations

Group	Name
DEFAULT	sqlite_synchronous

backend**Type** string**Default** `sqlalchemy`

The back end to use for the database.

Table 7: Deprecated Variations

Group	Name
DEFAULT	db_backend

connection**Type** string**Default** <None>

The SQLAlchemy connection string to use to connect to the database.

Table 8: 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.

Table 9: Deprecated Variations

Group	Name
DATABASE	idle_timeout
database	idle_timeout
DEFAULT	sql_idle_timeout
DATABASE	sql_idle_timeout
sql	idle_timeout

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.

Table 10: Deprecated Variations

Group	Name
DEFAULT	sql_max_pool_size
DATABASE	sql_max_pool_size

max_retries

Type integer

Default 10

Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Table 11: 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 12: 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 13: 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 14: Deprecated Variations

Group	Name
DEFAULT	sql_connection_debug

connection_trace

Type boolean

Default False

Add Python stack traces to SQL as comment strings.

Table 15: 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 16: 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&

glance

Configuration options for the Image service

num_retries**Type** integer**Default** 0**Minimum Value** 0

Enable glance operation retries.

Specifies the number of retries when uploading / downloading an image to / from glance. 0 means no retries.

verify_glance_signatures**Type** boolean**Default** False

Enable image signature verification.

cyborg uses the image signature metadata from glance and verifies the signature of a signed image while downloading that image. If the image signature cannot be verified or if the image signature metadata is either incomplete or unavailable, then cyborg will not boot the image and instead will place the instance into an error state. This provides end users with stronger assurances of the integrity of the image data they are using to create servers.

Related options:

- The options in the *key_manager* group, as the *key_manager* is used for the signature validation.
- Both *enable_certificate_validation* and *default_trusted_certificate_ids* below depend on this option being enabled.

enable_certificate_validation**Type** boolean**Default** False

Enable certificate validation for image signature verification.

During image signature verification cyborg will first verify the validity of the images signing certificate using the set of trusted certificates associated with the instance. If certificate validation fails, signature verification will not be performed and the instance will be placed into an error state. This provides end users with stronger assurances that the image data is unmodified and trustworthy. If left disabled, image signature verification can still occur but the end user will

not have any assurance that the signing certificate used to generate the image signature is still trustworthy.

Related options:

- This option only takes effect if `verify_glance_signatures` is enabled.
- The value of `default_trusted_certificate_ids` may be used when this option is enabled.

Warning: This option is deprecated for removal since 16.0.0. Its value may be silently ignored in the future.

Reason This option is intended to ease the transition for deployments leveraging image signature verification. The intended state long-term is for signature verification and certificate validation to always happen together.

`default_trusted_certificate_ids`

Type list

Default []

List of certificate IDs for certificates that should be trusted.

May be used as a default list of trusted certificate IDs for certificate validation. The value of this option will be ignored if the user provides a list of trusted certificate IDs with an instance API request. The value of this option will be persisted with the instance data if signature verification and certificate validation are enabled and if the user did not provide an alternative list. If left empty when certificate validation is enabled the user must provide a list of trusted certificate IDs otherwise certificate validation will fail.

Related options:

- The value of this option may be used if both `verify_glance_signatures` and `enable_certificate_validation` are enabled.

`debug`

Type boolean

Default False

Enable or disable debug logging with glanceclient.

`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

`keyfile`

Type string

Default <None>

PEM encoded client certificate key file

insecure

Type boolean

Default False

Verify HTTPS connections.

timeout

Type integer

Default <None>

Timeout value for http requests

collect_timing

Type boolean

Default False

Collect per-API call timing information.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

service_type

Type string

Default image

The default service_type for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

region_name

Type string

Default <None>

The default `region_name` for endpoint URL discovery.

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.

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.

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.

keystone

Configuration options for the identity service

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

keyfile

Type string

Default <None>

PEM encoded client certificate key file

insecure

Type boolean

Default False

Verify HTTPS connections.

timeout

Type integer

Default <None>

Timeout value for http requests

collect_timing

Type boolean

Default False

Collect per-API call timing information.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

service_type

Type string

Default identity

The default service_type for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

region_name

Type string

Default <None>

The default `region_name` for endpoint URL discovery.

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.

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.

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.

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 17: 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 you're 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 18: 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 False

(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.

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 19: Deprecated Variations

Group	Name
keystone_authtoken	auth_plugin

auth_section

Type unknown type

Default <None>

Config Section from which to load plugin specific options

nova

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

keyfile

Type string

Default <None>

PEM encoded client certificate key file

insecure

Type boolean

Default `False`

Verify HTTPS connections.

timeout

Type `integer`

Default `<None>`

Timeout value for http requests

collect_timing

Type `boolean`

Default `False`

Collect per-API call timing information.

split_loggers

Type `boolean`

Default `False`

Log requests to multiple loggers.

auth_type

Type `unknown type`

Default `<None>`

Authentication type to load

Table 20: Deprecated Variations

Group	Name
nova	auth_plugin

auth_section

Type `unknown type`

Default `<None>`

Config Section from which to load plugin specific options

auth_url

Type `unknown type`

Default `<None>`

Authentication URL

system_scope

Type `unknown type`

Default `<None>`

Scope for system operations

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

project_id

Type unknown type

Default <None>

Project ID to scope to

project_name

Type unknown type

Default <None>

Project name to scope to

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

trust_id

Type unknown type

Default <None>

Trust ID

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.

user_id

Type unknown type

Default <None>

User ID

username

Type unknown type

Default <None>

Username

Table 21: Deprecated Variations

Group	Name
nova	user-name
nova	user_name

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

password

Type unknown type

Default <None>

Users password

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

service_type

Type string

Default `compute`

The default `service_type` for endpoint URL discovery.

service_name

Type string

Default `<None>`

The default `service_name` for endpoint URL discovery.

valid_interfaces

Type list

Default `['internal', 'public']`

List of interfaces, in order of preference, for endpoint URL.

region_name

Type string

Default `<None>`

The default `region_name` for endpoint URL discovery.

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.

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.

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 retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

oslo_messaging_amqp

container_name

Type string

Default <None>

Name for the AMQP container. must be globally unique. Defaults to a generated UUID

Table 22: Deprecated Variations

Group	Name
amqp1	container_name

idle_timeout

Type integer

Default 0

Timeout for inactive connections (in seconds)

Table 23: Deprecated Variations

Group	Name
amqp1	idle_timeout

trace

Type boolean

Default False

Debug: dump AMQP frames to stdout

Table 24: 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 25: Deprecated Variations

Group	Name
amqp1	ssl_ca_file

ssl_cert_file

Type string

Default ''

Self-identifying certificate PEM file for client authentication

Table 26: 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 27: Deprecated Variations

Group	Name
amqp1	ssl_key_file

ssl_key_password

Type string

Default <None>

Password for decrypting ssl_key_file (if encrypted)

Table 28: 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 29: Deprecated Variations

Group	Name
amqp1	sasl_mechanisms

sasl_config_dir

Type string

Default ''

Path to directory that contains the SASL configuration

Table 30: Deprecated Variations

Group	Name
amqp1	sasl_config_dir

sasl_config_name

Type string

Default ''

Name of configuration file (without `.conf` suffix)

Table 31: 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 32: Deprecated Variations

Group	Name
amqp1	server_request_prefix

broadcast_prefix

Type string

Default broadcast

address prefix used when broadcasting to all servers

Table 33: Deprecated Variations

Group	Name
amqp1	broadcast_prefix

group_request_prefix**Type** string**Default** unicast

address prefix when sending to any server in group

Table 34: 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 35: 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 36: Deprecated Variations

Group	Name
DEFAULT	notification_transport_url

topics

Type list

Default ['notifications']

AMQP topic used for OpenStack notifications.

Table 37: 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 38: Deprecated Variations

Group	Name
DEFAULT	amqp_auto_delete

ssl**Type** boolean**Default** False

Connect over SSL.

Table 39: 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 40: 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 41: 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 42: 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 43: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_ca_certs

heartbeat_in_pthread

Type boolean

Default False

EXPERIMENTAL: Run the health check heartbeat thread through a native python thread. By default if this option isnt provided the health check heartbeat will inherit the execution model from the parent process. By example if the parent process have monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.

kombu_reconnect_delay

Type floating point

Default 1.0

How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Table 44: 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 45: 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 46: 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 47: 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 48: 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 let 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_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.json`

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 49: Deprecated Variations

Group	Name
DEFAULT	policy_file

policy_default_rule

Type string

Default `default`

Default rule. Enforced when a requested rule is not found.

Table 50: Deprecated Variations

Group	Name
DEFAULT	policy_default_rule

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 51: 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

placement

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

keyfile

Type string

Default <None>

PEM encoded client certificate key file

insecure

Type boolean

Default False

Verify HTTPS connections.

timeout

Type integer

Default <None>

Timeout value for http requests

collect_timing

Type boolean

Default False

Collect per-API call timing information.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 52: Deprecated Variations

Group	Name
placement	auth_plugin

auth_section

Type unknown type

Default <None>

Config Section from which to load plugin specific options

auth_url

Type unknown type

Default <None>

Authentication URL

system_scope

Type unknown type

Default <None>

Scope for system operations

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

project_id

Type unknown type

Default <None>

Project ID to scope to

project_name

Type unknown type

Default <None>

Project name to scope to

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

trust_id

Type unknown type

Default <None>

Trust ID

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.

user_id

Type unknown type

Default <None>

User ID

username

Type unknown type

Default <None>

Username

Table 53: Deprecated Variations

Group	Name
placement	user-name
placement	user_name

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

password

Type unknown type

Default <None>

Users password

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

service_type

Type string

Default placement

The default service_type for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

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.

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.

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 retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

service_user

Configuration options for service to service authentication using a service token. These options allow sending a service token along with the users token when contacting external REST APIs.

send_service_user_token

Type boolean

Default `False`

When True, if sending a user token to a REST API, also send a service token.

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

keyfile

Type string

Default <None>

PEM encoded client certificate key file

insecure

Type boolean

Default `False`

Verify HTTPS connections.

timeout

Type integer

Default <None>

Timeout value for http requests

collect_timing

Type boolean

Default `False`

Collect per-API call timing information.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 54: Deprecated Variations

Group	Name
service_user	auth_plugin

auth_section

Type unknown type

Default <None>

Config Section from which to load plugin specific options

auth_url

Type unknown type

Default <None>

Authentication URL

system_scope

Type unknown type

Default <None>

Scope for system operations

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

project_id

Type unknown type

Default <None>

Project ID to scope to

project_name

Type unknown type

Default <None>

Project name to scope to

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

trust_id

Type unknown type

Default <None>

Trust ID

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.

user_id

Type unknown type

Default <None>

User ID

username

Type unknown type

Default <None>

Username

Table 55: Deprecated Variations

Group	Name
service_user	user-name
service_user	user_name

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

password

Type unknown type

Default <None>

Users password

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

ssl

ca_file

Type string

Default <None>

CA certificate file to use to verify connecting clients.

Table 56: Deprecated Variations

Group	Name
DEFAULT	ssl_ca_file

cert_file**Type** string**Default** <None>

Certificate file to use when starting the server securely.

Table 57: 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 58: 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.

Cyborg Configuration Sample

Sample configuration files for all Cyborg services are found in the online version of this documentation.

Cyborg Sample Policy

Warning: JSON formatted policy file is deprecated since Cyborg 5.0.0(Victoria). Use YAML formatted file. Use [oslopolicy-convert-json-to-yaml](#) tool to convert the existing JSON to YAML formatted policy file in backward compatible way.

The following is a sample cyborg policy file that has been auto-generated from default policy values in code. If you're using the default policies, then the maintenance of this file is not necessary, and it should

not be copied into a deployment. Doing so will result in duplicate policy definitions. It is here to help explain which policy operations protect specific cyborg APIs, but it is not suggested to copy and paste into a deployment unless you're planning on providing a different policy for an operation that is not the default.

If you wish build a policy file, you can also use `tox -e genpolicy` to generate it.

The sample policy file can also be downloaded in [file form](#).

```
# Default rule for System Admin APIs.
#"system_admin_api": "role:admin and system_scope:all"

# Default rule for System level read only APIs.
#"system_reader_api": "role:reader and system_scope:all"

# Default rule for Project level admin APIs.
#"project_admin_api": "role:admin and project_id:%(project_id)s"

# Default rule for Project level non admin APIs.
#"project_member_api": "role:member and project_id:%(project_id)s"

# Default rule for Project level read only APIs.
#"project_reader_api": "role:reader and project_id:%(project_id)s"

# Default rule for system_admin+owner APIs.
#"system_admin_or_owner": "rule:system_admin_api or rule:project_member_api
→"

# Default rule for System+Project read only APIs.
#"system_or_project_reader": "rule:system_reader_api or rule:project_
→reader_api"

# DEPRECATED
# "public_api" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# legacy rule of Internal flag for public API routes
#"public_api": "is_public_api:True"

# DEPRECATED
# "allow" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# legacy rule: any access will be passed
#"allow": "@"

# DEPRECATED
# "deny" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
```

(continues on next page)

(continued from previous page)

```

# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# legacy rule: all access will be forbidden
#"deny": "!"

# DEPRECATED
# "default" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# Legacy rule for default rule
#"default": "rule:admin_or_owner"

# DEPRECATED
# "admin_api" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# Legacy rule for cloud admin access
#"admin_api": "role:admin or role:administrator"

# DEPRECATED
# "is_admin" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# Full read/write API access
#"is_admin": "rule:admin_api"

# DEPRECATED
# "admin_or_owner" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies
# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# Admin or owner API access
#"admin_or_owner": "is_admin:True or project_id:%(project_id)s"

# DEPRECATED
# "admin_or_user" has been deprecated since W.
# Cyborg API policies are introducing new default roles with
# scope_type capabilities. We will start to deprecate old policies

```

(continues on next page)

(continued from previous page)

```

# from WALLABY release, and are going to ignore all the old policies
# silently from X release. Be sure to take these new defaults into
# consideration if you are relying on overrides in your deployment for
# the policy API.
# Admin or user API access
#"admin_or_user": "is_admin:True or user_id:%(user_id)s"

# Retrieve all device_profiles
# GET /v2/device_profiles
# Intended scope(s): system, project
#"cyborg:device_profile:get_all": "rule:system_or_project_reader"

# DEPRECATED
# "cyborg:device_profile:get_all":"rule:admin_or_owner" has been
# deprecated since W in favor of
# "cyborg:device_profile:get_all":"rule:system_or_project_reader".
# request admin_or_owner rule is too strict for listing device_profile

# Retrieve a specific device_profile
# GET /v2/device_profiles/{device_profiles_uuid}
# Intended scope(s): system, project
#"cyborg:device_profile:get_one": "rule:system_or_project_reader"

# DEPRECATED
# "cyborg:device_profile:get_one":"rule:admin_or_owner" has been
# deprecated since W in favor of
# "cyborg:device_profile:get_one":"rule:system_or_project_reader".
# request admin_or_owner rule is too strict for retrieving a
# device_profile

# Create a device_profile
# POST /v2/device_profiles
# Intended scope(s): system
#"cyborg:device_profile:create": "rule:system_admin_api"

# DEPRECATED
# "cyborg:device_profile:create":"rule:is_admin" has been deprecated
# since W in favor of
# "cyborg:device_profile:create":"rule:system_admin_api".
# project_admin_or_owner is too permissive, introduce system_scoped
# admin for creation

# Delete device_profile(s)
# DELETE /v2/device_profiles/{device_profiles_uuid}
# DELETE /v2/device_profiles?value={device_profile_name1}
# Intended scope(s): system
#"cyborg:device_profile:delete": "rule:system_admin_api"

# DEPRECATED
# "cyborg:device_profile:delete":"rule:admin_or_owner" has been
# deprecated since W in favor of
# "cyborg:device_profile:delete":"rule:system_admin_api".
# project_admin_or_owner is too permissive, introduce system_scoped
# admin for deletion

# Show device detail

```

(continues on next page)

(continued from previous page)

```
#"cyborg:device:get_one": "rule:allow"

# Retrieve all device records
#"cyborg:device:get_all": "rule:allow"

# Show deployable detail
#"cyborg:deployable:get_one": "rule:allow"

# Retrieve all deployable records
#"cyborg:deployable:get_all": "rule:allow"

# FPGA programming.
#"cyborg:deployable:program": "rule:allow"

# Retrieve accelerator request records.
#"cyborg:arq:get_all": "rule:default"

# Get an accelerator request record.
#"cyborg:arq:get_one": "rule:default"

# Create accelerator request records.
#"cyborg:arq:create": "rule:allow"

# Delete accelerator request records.
#"cyborg:arq:delete": "rule:default"

# Update accelerator request records.
#"cyborg:arq:update": "rule:default"

# Show fpga detail
#"cyborg:fpga:get_one": "rule:allow"

# Retrieve all fpga records
#"cyborg:fpga:get_all": "rule:allow"

# Update fpga records
#"cyborg:fpga:update": "rule:allow"
```

Cyborg Policy Configuration Guide

Cyborg, like most OpenStack projects, uses a policy language to restrict permissions on REST API actions.

- *Policy Concepts*: In the Victoria release, Cyborg API policy defines new default roles with system scope capabilities. These new changes improve the security level and manageability of Cyborg API as they are richer in terms of handling access at system and project level token with Read and Write roles.

Understanding Cyborg Policies

Warning: JSON formatted policy file is deprecated since Cyborg (Victoria). Use YAML formatted file. Use `oslopolicy-convert-json-to-yaml` tool to convert the existing JSON to YAML formatted policy file in backward compatible way.

Cyborg supports a rich policy system that has evolved significantly over its lifetime. Initially, cyborg policy defaults have been defined in the codebase, requiring the `policy.json` file only to override these defaults. Starting in the Victoria release, policy file has been changed from `policy.json` to `policy.yaml`.

The old default policy in Cyborg is incomplete and not good enough. Since Cyborg V2 API is newly implemented in Train, RBAC check for V2 API still remains incomplete. So in the Ussuri release, the specification of policy refresh was approved. In the Victoria release, Cyborg landed the new default roles to improve some issues that had been identified:

1. No `allow`. Old policy `allow` means any access will be passed. `allow` rule was used by `cyborg:arq:create`, which is too slack.
2. No global vs project admin. The old role `is_admin` is used for the global admin that is able to make almost any change to Cyborg, and see all details of the Cyborg system. The rule passes for any user with an admin role, it doesnt matter which project is used.
3. No `admin_or_owner`. Old role `admin_or_owner` sounds like it checks if the user is a member of a project. However, for most APIs we use the default target which means this rule will pass for any authenticated user.
4. Introduce `scope_type` and `reader` role. There still some cases which are not well covered. For example, it is impossible to allow a user to retrieve/update devices which are shared by multiple projects from a system level without being given the global admin role. In addition, cyborg now doesnt have a `reader` role.

Keystone comes with `admin`, `member` and `reader` roles by default. Please refer to [keystone document](#) for more information about these new defaults. In addition, keystone supports a new system scope concept that makes it easier to protect deployment level resources from project or system level resources. Please refer to [token scopes](#) and [system scope specification](#) to understand the scope concept.

In the Cyborg (Victoria) release, Cyborg policies implemented the scope concept and default roles provided by keystone (`admin`, `member`, and `reader`). Using common roles from keystone reduces the likelihood of similar, but different, roles implemented across projects or deployments. With the help of the new defaults it is easier to understand who can do what across projects, reduces divergence, and increases interoperability.

The below sections explain how these new defaults in the Cyborg can solve the issues mentioned above and extend more functionality to end users in a safe and secure way.

More information is provided in the [cyborg specification](#)

Scope

OpenStack Keystone supports different scopes in tokens. These are described [here](#). Token scopes represent the layer of authorization. Policy `scope_types` represent the layer of authorization required to access an API.

Note: The `scope_type` of each policy is hardcoded and is not overridable via the policy file.

Cyborg policies have implemented the scope concept by defining the `scope_type` in policies. To know each policy's `scope_type`, please refer to the [Policy Reference](#) and look for Scope Types or Intended scope(s) in [Policy Sample File](#) as shown in below examples.

system scope

Policies with a `scope_type` of `system` means a user with a `system-scoped` token has permission to access the resource. This can be seen as a global role. All the system-level operations policies have defaulted to `scope_type` of `['system']`.

For example, consider the `POST /v2/device_profiles` API.

```
# Create a device_profile
# POST /v2/device_profiles
# Intended scope(s): system
#"cyborg:device_profile:create": "rule:system_admin_api"
```

project scope

Policies with a `scope_type` of `project` means a user with a `project-scoped` token has permission to access the resource. This can be seen as a project role. All the project-level operations policies should be set to `scope_type` of `['project']` by default.

system and project scope

Policies with a `scope_type` of `system` and `project` means a user with a `system-scoped` or `project-scoped` token has permission to access the resource. All the system and project level operations policies have defaulted to `scope_type` of `['system', 'project']`.

For example, consider the `GET /v2/device_profiles/{device_profiles_uuid}` API.

```
# Retrieve a specific device_profile
# GET /v2/device_profiles/{device_profiles_uuid}
# Intended scope(s): system, project
#"cyborg:device_profile:get_one": "rule:system_or_project_reader"
```

These scope types provide a way to differentiate between system-level and project-level access roles. You can control the information with scope of the users.

Policy scope is disabled by default to allow operators to migrate from the old policy enforcement system in a graceful way. This can be enabled by configuring the `oslo_policy.enforce_scope` option to `True`.

Note: [oslo_policy] enforce_scope=True

Roles

You can refer to [this](#) document to know about all available defaults from Keystone.

Along with the `scope_type` feature, Cyborg policy defines new defaults for each policy.

reader

This provides read-only access to the resources within the `system` or `project`. Cyborg policies are defaulted to below rules:

```
system_reader_api
  Default
    role:reader and system_scope:all

project_reader_api
  Default
    role:reader and project_id:%(project_id)s

system_or_project_reader
  Default
    rule:system_reader_api or rule:project_reader_api
```

member

This role is to perform the project level write operation with combination to the system admin. Cyborg policies are defaulted to below rules:

```
project_member_api
  Default
    role:member and project_id:%(project_id)s

system_admin_or_owner
  Default
    rule:system_admin_api or rule:project_member_api
```

admin

This role is to perform the admin level write operation at system as well as at project-level operations. Cyborg policies are defaulted to below rules:

```
system_admin_api
  Default
    role:admin and system_scope:all

project_admin_api
  Default
```

(continues on next page)

(continued from previous page)

```

    role:admin and project_id:%(project_id)s
system_admin_or_owner
    Default
    rule:system_admin_api or rule:project_member_api

```

With these new defaults, you can solve the problem of:

1. Providing the read-only access to the user. Policies are made more granular and defaulted to reader rules. For example: If you need to let someone audit your deployment for security purposes.
2. Customize the policy in better way. For example, you will be able to provide access to project level member to perform arq patch/post for instance boot with the projects token.

Backward Compatibility

During the development period (Victoria and Wallaby releases), the new and old policy will both work for backward compatibility by supporting the old defaults and disabling the `scope_type` feature by default. This means the old defaults and deployments that use them will keep working as-is. However, we encourage every deployment to switch to new policy. `scope_type` will be enabled by default and the old defaults will be removed starting in the X release.

To implement the new default reader roles, some policies needed to become granular. They have been renamed, with the old names still supported for backwards compatibility.

Migration Plan

To have a graceful migration, Cyborg provides two flags to switch to the new policy completely. You do not need to overwrite the policy file to adopt the new policy defaults.

Here is step wise guide for migration:

1. Create scoped token:

You need to create the new token with scope knowledge via below CLI:

- [Create System Scoped Token](#).
- [Create Project Scoped Token](#).

2. Create new default roles in keystone if not done:

If you do not have new defaults in Keystone then you can create and re-run the [Keystone Bootstrap](#). Keystone added this support in 14.0.0 (Rocky) release.

3. Enable Scope Checks

The `oslo_policy.enforce_scope` flag is to enable the `scope_type` features. The scope of the token used in the request is always compared to the `scope_type` of the policy. If the scopes do not match, one of two things can happen. If `oslo_policy.enforce_scope` is True, the request will be rejected. If `oslo_policy.enforce_scope` is False, an warning will be logged, but the request will be accepted (assuming the rest of the policy passes). The default value of this flag is False.

Note: Before you enable this flag, you need to audit your users and make sure everyone who needs system-level access has a system role assignment in keystone.

4. Enable new defaults

The `oslo_policy.enforce_new_defaults` flag switches the policy to new defaults-only. This flag controls whether or not to use old deprecated defaults when evaluating policies. If True, the old deprecated defaults are not evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be rejected. The default value of this flag is False.

Note: Before you enable this flag, you need to educate users about the different roles they need to use to continue using Cyborg APIs.

5. Check for deprecated policies

A few policies were made more granular to implement the reader roles. New policy names are available to use. If old policy names which are renamed are overwritten in policy file, then warning will be logged. Please migrate those policies to new policy names.

We expect all deployments to migrate to new policy by X release so that we can remove the support of old policies.

- *Policy Reference*: A complete reference of all policy points in cyborg and what they impact.

Cyborg Policies

The following is an overview of all available policies in Cyborg.

Warning: JSON formatted policy file is deprecated since Cyborg (Victoria). Use YAML formatted file. Use `oslopolicy-convert-json-to-yaml` tool to convert the existing JSON to YAML formatted policy file in backward compatible way.

cyborg.api

`system_admin_api`

Default `role:admin and system_scope:all`

Default rule for System Admin APIs.

`system_reader_api`

Default `role:reader and system_scope:all`

Default rule for System level read only APIs.

`project_admin_api`

Default `role:admin and project_id:%(project_id)s`

Default rule for Project level admin APIs.

project_member_api

Default `role:member and project_id:%(project_id)s`

Default rule for Project level non admin APIs.

project_reader_api

Default `role:reader and project_id:%(project_id)s`

Default rule for Project level read only APIs.

system_admin_or_owner

Default `rule:system_admin_api or rule:project_member_api`

Default rule for system_admin+owner APIs.

system_or_project_reader

Default `rule:system_reader_api or rule:project_reader_api`

Default rule for System+Project read only APIs.

public_api

Default `is_public_api:True`

legacy rule of Internal flag for public API routes

allow

Default `@`

legacy rule: any access will be passed

deny

Default `!`

legacy rule: all access will be forbidden

default

Default `rule:admin_or_owner`

Legacy rule for default rule

admin_api

Default `role:admin or role:administrator`

Legacy rule for cloud admin access

is_admin

Default `rule:admin_api`

Full read/write API access

admin_or_owner

Default `is_admin:True or project_id:%(project_id)s`

Admin or owner API access

admin_or_user

Default `is_admin:True` or `user_id:%(user_id)s`

Admin or user API access

cyborg:device_profile:get_all

Default `rule:system_or_project_reader`

Operations

- **GET** `/v2/device_profiles`

Scope Types

- **system**
- **project**

Retrieve all device_profiles

cyborg:device_profile:get_one

Default `rule:system_or_project_reader`

Operations

- **GET** `/v2/device_profiles/{device_profiles_uuid}`

Scope Types

- **system**
- **project**

Retrieve a specific device_profile

cyborg:device_profile:create

Default `rule:system_admin_api`

Operations

- **POST** `/v2/device_profiles`

Scope Types

- **system**

Create a device_profile

cyborg:device_profile:delete

Default `rule:system_admin_api`

Operations

- **DELETE** `/v2/device_profiles/{device_profiles_uuid}`
- **DELETE** `/v2/device_profiles?value={device_profile_name1}`

Scope Types

- **system**

Delete device_profile(s)

cyborg:device:get_one

Default rule:allow

Show device detail

cyborg:device:get_all

Default rule:allow

Retrieve all device records

cyborg:deployable:get_one

Default rule:allow

Show deployable detail

cyborg:deployable:get_all

Default rule:allow

Retrieve all deployable records

cyborg:deployable:program

Default rule:allow

FPGA programming.

cyborg:arq:get_all

Default rule:default

Retrieve accelerator request records.

cyborg:arq:get_one

Default rule:default

Get an accelerator request record.

cyborg:arq:create

Default rule:allow

Create accelerator request records.

cyborg:arq:delete

Default rule:default

Delete accelerator request records.

cyborg:arq:update

Default rule:default

Update accelerator request records.

cyborg:fpga:get_one

Default rule:allow

Show fpga detail

cyborg:fpga:get_all

Default rule:allow

Retrieve all fpga records

cyborg:fpga:update

Default `rule:allow`

Update fpga records

2.2.2 Cyborg Support Matrix

Cyborg supports specific operations on VMs with attached accelerator resources, which are generally a subset of the full set of VM operations supported by Nova ([nova-vm-ops](#)).

In this release, these operations have a dependency on specific Nova patches ([nova-patches](#)). They can be expected to work in Cyborg only if and when these Nova patches get merged without significant changes. These operations are not supported in this release since the dependencies are not met.

Table 59: VM Operations Expected to Work With Nova Dependencies

VM Operation	Command
VM creation	<code>openstack server create</code>
VM deletion	<code>openstack server delete</code>
Reboot within VM	<code>ssh to VM and reboot in OS</code>
Soft reboot	<code>openstack server reboot --soft</code>
Pause/Unpause	<code>openstack server pause, openstack server unpause</code>
Lock/Unlock	<code>openstack server lock, openstack server unlock</code>

Operations not listed here may or may not work.

Driver Support

The list of drivers available as part of the Cyborg distribution at the time of release can be found in: `cyborg.accelerator.driver` section of [Cyborgs setup.cfg](#)

The following table provides additional information for individual drivers.

Table 60: Driver Support

Driver Name	Supported Products	Description	Notes	Temporary Test Report
Fake Driver	None	A driver that creates a fake device with accelerator resources of type FPGA. Useful for exploring Cyborg without hardware and for Continuous Integration testing.	None	None
Intel FPGA OPAE Driver	Intel PAC	The driver for Intel FPGA devices with OPAE software stack.	Supports programming of FPGA bitstreams of type gbs.	None
Nvidia GPU driver	None	The driver for Nvidia GPUs.	None	None
Ascend AI Chip driver	None	The driver for Huawei's Ascend AI chips.	None	None
Intel QAT Driver	Intel Quick-Assist Technology Card	The driver for Intel QAT Cards.	None	Test results reported at Aug 2020. Please reference: Intel QAT Driver Test Report
Inspur FPGA Driver	None	The driver for Inspur FPGA Cards.	None	Test results reported at Aug 2020. Please reference: Inspur FPGA Driver Test Report

Note: Temporary Test Report: This is a temporary test report, it is only valid for a short time, if you encounter problems, please contact the [Cyborg team](#).

FOR END USERS

As an end user of Cyborg, you'll use Cyborg to create and manage accelerators with either tools or the API directly.

3.1 Tools for using Cyborg

Information on the commands available through Cyborg's Command Line Interface (CLI) can be found in this section of documentation.

3.1.1 Command-Line Interface Reference

cyborg-status

Synopsis

```
cyborg-status <category> <command> [<args>]
```

Description

cyborg-status is a tool that provides routines for checking the status of a Cyborg deployment.

Options

The standard pattern for executing a **cyborg-status** command is:

```
cyborg-status <category> <command> [<args>]
```

Run without arguments to see a list of available command categories:

```
cyborg-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:

```
cyborg-status upgrade
```

These sections describe the available categories and arguments for `cyborg-status`.

Upgrade

cyborg-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

2.0.0 (Stein)

- Placeholder to be filled in with checks as they are added in Stein.

3.2 Using the API

Following the Ussuri release, every Cyborg deployment should have the following endpoints:

`/` - list of available versions

`/v2` - the version 2 of the Acceleration API, it uses microversions

`/v2.0` - same API as v2, except uses microversions

The following guide concentrates on documenting the v2 API, please note that the v2.0 is the first microversion of the v2 API and are also covered by this guide.

- [Cyborg API Reference](#): The complete reference for the accelerator API, including all methods and request / response parameters and their meaning.

DOCUMENTATION FOR DEVELOPERS

4.1 So You Want to Contribute

For general information on contributing to OpenStack, please check out the [contributor guide](#) to get started. It covers all the basics that are common to all OpenStack projects: the accounts you need, the basics of interacting with our Gerrit review system, how we communicate as a community, etc.

Below will cover the more project specific information you need to get started with `{{cookiecutter.service}}`.

4.1.1 Communication

We use the `#openstack-cyborg` IRC channel.

The weekly meetings happen in this channel. You can find the meeting times, previous meeting logs and proposed meeting agendas at [Cyborg Team Meeting Page](#).

4.1.2 Contacting the Core Team

The core reviewers of Cyborg and their emails are listed in [Cyborg core team](#).

4.1.3 New Feature Planning

To propose or plan new features, we add a new story in the [Cyborg Storyboard](#) and/or propose a specification in the [cyborg-specs](#) repository.

4.1.4 Task Tracking

We track our tasks in the [Cyborg Storyboard](#).

If you're looking for some smaller, easier work item to pick up and get started on, ask in the IRC meeting.

4.1.5 Reporting a Bug

You found an issue and want to make sure we are aware of it? You can do so by adding an entry in the [Cyborg Storyboard](#) or raising it in the IRC meeting.

4.1.6 Getting Your Patch Merged

To merge a patch, it must pass all voting Zuul checks and get two +2s from core reviewers. We strive to avoid scenarios where one person from a company or organization proposes a patch, and two other core reviewers from the same organization approve it to get it merged. In other words, at least one among the patch author and the two approving reviewers must be from another organization.

We are constantly striving to improve quality. Proposed patches must generally have unit tests and/or functional tests that cover the changes, and strive to improve code coverage.

4.1.7 Project Team Lead Duties

All common PTL duties are enumerated in the [PTL guide](#).

4.2 DevStack Quick Start

4.2.1 Create stack user (optional)

Devstack should be run as a non-root user with sudo enabled (standard logins to cloud images such as ubuntu or cloud-user are usually fine).

You can quickly create a separate stack user to run DevStack with.

```
$ sudo useradd -s /bin/bash -d /opt/stack -m stack
```

Since this user will be making many changes to your system, it should have sudo privileges:

```
$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
```

```
$ sudo su - stack
```

4.2.2 Download DevStack

```
$ git clone https://opendev.org/openstack/devstack
```

```
$ cd devstack
```

The *devstack* repo contains a script that installs OpenStack.

4.2.3 Create local.conf file

Create a *local.conf* file at the root of the devstack git repo.

Host settings

```
[[local|localrc]]
HOST_IP=YOUR_IP_CONFIG
SERVICE_HOST=$HOST_IP
DATABASE_TYPE=mysql
MYSQL_HOST=$HOST_IP
RABBIT_HOST=$HOST_IP
```

- Replace YOUR_IP_CONFIG with your host IP (e.g. 10.0.0.72 or localhost).

Password settings

```
# Passwords
DATABASE_PASSWORD=123
ADMIN_PASSWORD=123
MYSQL_PASSWORD=123
RABBIT_PASSWORD=123
SERVICE_PASSWORD=123
SERVICE_TOKEN=ADMIN
```

- Pre-set the passwords to prevent interactive prompts.

Enable services

```
#FIXED_RANGE=192.168.128.0/24
#IPV4_ADDRS_SAFE_TO_USE=192.168.128.0/24
#GIT_BASE=/opt/git
disable_service n-net
disable_service tempest
disable_service heat
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service neutron
enable_service n-novnc
enable_plugin cyborg https://opendev.org/openstack/cyborg
NOVA_VNC_ENABLED=True
NOVNC_PROXY_URL="http://$SERVICE_HOST:6080/vnc_auto.html"
VNC_SERVER_LISTEN=0.0.0.0
VNC_SERVER_PROXYCLIENT_ADDRESS=$SERVICE_HOST
RECLONE=False
#enable Logging
LOGFILE=/opt/stack/logs/stack.sh.log
VERBOSE=True
LOG_COLOR=True
LOGDIR=/opt/stack/logs
```

- Uncomment GIT_BASE configuration if you have a local git repo
- enable_plugin cyborg will execute cyborg/devstack/plugin.sh and start cyborg service
- The devstack logs will appear in \$LOGDIR

Note: If you got version conflicts, please set *PIP_UPGRADE* to *True* in local.conf

4.2.4 Multi-Node Lab

If you want to setup an OpenStack with cyborg in a realistic test configuration with multiple physical servers. Please ref¹.

Cluster Controller

```
disable_service cyborg-agent
```

Compute Nodes

```
enable_service cyborg-agent  
disable_service cyborg-api  
disable_service cyborg-cond
```

- If you do not want to setup cyborg-agent on controller, you can disable it.
- You do not need to enable cyborg-api and cyborg-cond on compute nodes.

Testing with unmerged changes

To test with changes that have not been merged, the enable_plugin line can be modified to specify the branch/reference to be cloned.

```
enable_plugin cyborg https://review.opendev.org/openstack/cyborg refs/  
↪changes/28/708728/1
```

the format is

```
enable_plugin <directory name> <git repo url> <change/revision>
```

¹ Openstack Multi-Node Lab Setup

Cell V2 Deployment

Compute node services must be mapped to a cell before they can be used. Cell V2 deployment, please [ref²](#).

4.2.5 Run DevStack

```
$ ./stack.sh
```

This will take a 30-40 minutes, largely depending on the speed of your internet connection. Many git trees and packages will be installed during this process.

It will speed up your installation if you have a local GIT_BASE.

4.2.6 Use OpenStack

Command line

You can *source openrc YOUR_USER YOUR_USER* (e.g. *source openrc admin admin*) in your shell, and then use the *openstack* command line tool to manage your devstack.

Horizon

You can access horizon to experience the web interface to OpenStack, and manage vms, networks, volumes, and images from there.

References

4.3 Driver Development Guide

The goal of this document is to explain how to develop a new kind of Cyborg accelerator driver.

Note: Make sure you have installed Openstack environment using `devstack` before development.

4.3.1 Derive a new driver class

Imply the necessary interface, the list of interfaces is as follows:

```
class NewCyborgDriver(object):
    """Cyborg new accelerator driver.
    """

    def discover(self):
        """Discover specific accelerator
```

(continues on next page)

² Openstack Cell V2 Deployment Guide

(continued from previous page)

```
:return: list of cyborg.objects.driver_objects.driver_device.  
        DriverDevice  
"""  
pass
```

4.3.2 Modify setup.cfg

Add the new driver map into file `cyborg/setup.cfg`:

```
[entry_points]  
cyborg.accelerator.driver =  
    intel_fpga_driver = cyborg.accelerator.drivers.fpga.intel.  
→driver:IntelFPGADriver  
    new_driver_name = cyborg.accelerator.drivers.example.  
→driver:NewCyborgDriver
```

4.3.3 Reinstall and Test

Reinstall cyborg:

```
$ python setup.py develop
```

Restart cyborg-agent:

```
$ sudo systemctl restart devstack@cyborg-agent
```

INDICES AND TABLES

- search