

OpenStack

Configuration Reference

icehouse (June 1, 2015)

Icehouse



OpenStack Configuration Reference

icehouse (2015-06-01)

Copyright © 2013 OpenStack Foundation All rights reserved.

This document is for system administrators who want to look up configuration options. It contains lists of configuration options available with OpenStack and uses auto-generation to generate options and the descriptions from the code for each project. It includes sample configuration files.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

OpenStack configuration overview	16
Conventions	16
Document change history	17
Configuration file format	17
1. Block Storage	1
Introduction to the Block Storage service	1
cinder.conf configuration file	2
Volume drivers	3
Backup drivers	77
Block Storage sample configuration files	79
Log files used by Block Storage	121
Fibre Channel Zone Manager	121
Additional options	123
New, updated and deprecated options for cinder	136
2. Compute	140
Overview of nova.conf	140
Configure logging	141
Configure authentication and authorization	141
Configure resize	142
Database configuration	142
Configure the Oslo RPC messaging system	142
Configure the Compute API	146
Configure the EC2 API	148
Fibre Channel support in Compute	148
Hypervisors	148
Scheduling	180
Cells	195
Conductor	199
Example nova.conf configuration files	199
Compute log files	204
Compute sample configuration files	205
New, updated and deprecated options for nova	244
3. Dashboard	253
Configure the dashboard	253
Customize the dashboard	257
Additional sample configuration files	258
Log files used by the dashboard	269
4. Database Service	271
Configure the database	278
Configure the RPC messaging system	280
5. Identity service	284
Identity service configuration file	284
Identity service sample configuration files	298
New, updated and deprecated options for keystone	326
6. Image Service	330
Configure the API	336
Configure the RPC messaging system	338
Support for ISO images	340

Configuring Backends	341
Image Service sample configuration files	346
New, updated and deprecated options for glance	365
7. Networking	368
Networking configuration options	368
Log files used by Networking	400
Networking sample configuration files	401
New, updated and deprecated options for neutron	418
8. Object Storage	425
Introduction to Object Storage	425
Object Storage general service configuration	425
Object server configuration	427
Object expirer configuration	435
Container server configuration	438
Account server configuration	444
Proxy server configuration	450
Proxy server memcache configuration	466
Rsyncd configuration	466
Configure Object Storage features	467
9. Orchestration	483
Configure APIs	487
Configure Clients	490
Configure the RPC messaging system	493
New, updated and deprecated options for heat	496
10. Telemetry	500
Telemetry sample configuration files	509
New, updated and deprecated options for ceilometer	529
A. Firewalls and default ports	531
B. Community support	533
Documentation	533
ask.openstack.org	534
OpenStack mailing lists	534
The OpenStack wiki	535
The Launchpad Bugs area	535
The OpenStack IRC channel	536
Documentation feedback	536
OpenStack distribution packages	536

List of Figures

1.1. Ceph architecture	3
1.2. Repository Creation Plan screen	9
1.3. Local configuration	73
1.4. Remote configuration	73
2.1. VMware driver architecture	164
2.2. Filtering	181
2.3. Weighing hosts	190
2.4. KVM, Flat, MySQL, and Glance, OpenStack or EC2 API	203
2.5. KVM, Flat, MySQL, and Glance, OpenStack or EC2 API	204

List of Tables

1.1. Description of configuration options for storage_ceph	5
1.2. Description of configuration options for coraid	9
1.3. Description of configuration options for eqlx	11
1.4. Description of configuration options for storage_glusterfs	19
1.5. Description of configuration options for hds-hus	21
1.6. Configuration options	24
1.7. Huawei storage driver configuration options	37
1.8. Description of configuration options for xiv	38
1.9. Description of configuration options for storage_gpfs	39
1.10. Volume Create Options for GPFS Volume Drive	40
1.11. List of configuration flags for Storwize storage and SVC driver	43
1.12. Description of configuration options for storwize	44
1.13. Description of configuration options for lvm	48
1.14. Description of configuration options for netapp_cdot_iscsi	49
1.15. Description of configuration options for netapp_cdot_nfs	50
1.16. Description of extra specs options for NetApp Unified Driver with Clustered Data ONTAP	53
1.17. Description of configuration options for netapp_7mode_iscsi	55
1.18. Description of configuration options for netapp_7mode_nfs	56
1.19. Description of configuration options for netapp_eseries_iscsi	58
1.20. Description of configuration options for nexenta_iscsi	61
1.21. Description of configuration options for nexenta_nfs	63
1.22. Description of configuration options for storage_nfs	63
1.23. Description of configuration options for solidfire	66
1.24. Description of configuration options for vmware	67
1.25. Extra spec entry to VMDK disk file type mapping	68
1.26. Extra spec entry to clone type mapping	68
1.27. Description of configuration options for windows	72
1.28. Description of configuration options for storage_xen	74
1.29. Description of configuration options for zadara	76
1.30. Description of configuration options for backups_ceph	78
1.31. Description of configuration options for backups_tsm	78
1.32. Description of configuration options for backups_swift	79
1.33. Log files used by Block Storage services	121
1.34. Description of configuration options for zoning	121
1.35. Description of configuration options for zoning_manager	122
1.36. Description of configuration options for zoning_fabric	122
1.37. Description of configuration options for auth_token	123
1.38. Description of configuration options for connection	124
1.39. Description of configuration options for huawei	124
1.40. Description of configuration options for nas	124
1.41. Description of configuration options for hpmsa	125
1.42. Description of configuration options for database	125
1.43. Description of configuration options for keymgr	125
1.44. Description of configuration options for storage	125
1.45. Description of configuration options for rpc	126
1.46. Description of configuration options for san-solaris	128
1.47. Description of configuration options for rootwrap	128

2.33. Description of configuration options for libvirt	217
2.34. Description of configuration options for livemigration	218
2.35. Description of configuration options for logging	219
2.36. Description of configuration options for metadata	220
2.37. Description of configuration options for network	220
2.38. Description of configuration options for neutron	222
2.39. Description of configuration options for pci	223
2.40. Description of configuration options for periodic	223
2.41. Description of configuration options for policy	223
2.42. Description of configuration options for quota	224
2.43. Description of configuration options for rdp	224
2.44. Description of configuration options for redis	224
2.45. Description of configuration options for rootwrap	225
2.46. Description of configuration options for rpc_all	225
2.47. Description of configuration options for s3	227
2.48. Description of configuration options for scheduling	227
2.49. Description of configuration options for spice	229
2.50. Description of configuration options for testing	229
2.51. Description of configuration options for tilera	229
2.52. Description of configuration options for trustedcomputing	230
2.53. Description of configuration options for upgrade_levels	230
2.54. Description of configuration options for vmware	230
2.55. Description of configuration options for vnc	231
2.56. Description of configuration options for volumes	231
2.57. Description of configuration options for vpn	232
2.58. Description of configuration options for wsgi	233
2.59. Description of configuration options for xen	233
2.60. Description of configuration options for xvpnvncproxy	235
2.61. Description of configuration options for zookeeper	235
2.62. New options	244
2.63. New default values	250
2.64. Deprecated options	250
3.1. Log files used by the dashboard/httpd	270
4.1. Description of configuration options for api	271
4.2. Description of configuration options for auth_token	271
4.3. Description of configuration options for backup	273
4.4. Description of configuration options for common	273
4.5. Description of configuration options for compute	274
4.6. Description of configuration options for debug	274
4.7. Description of configuration options for dns	275
4.8. Description of configuration options for guestagent	275
4.9. Description of configuration options for heat	276
4.10. Description of configuration options for logging	276
4.11. Description of configuration options for nova	277
4.12. Description of configuration options for quota	277
4.13. Description of configuration options for redis	277
4.14. Description of configuration options for ssl	277
4.15. Description of configuration options for swift	277
4.16. Description of configuration options for taskmanager	277
4.17. Description of configuration options for volume	278
4.18. Description of configuration options for database	278

4.19. Description of configuration options for db_cassandra	279
4.20. Description of configuration options for db_couchbase	279
4.21. Description of configuration options for db_mongodb	279
4.22. Description of configuration options for db_mysql	279
4.23. Description of configuration options for db_percona	280
4.24. Description of configuration options for db_redis	280
4.25. Description of configuration options for rabbitmq	281
4.26. Description of configuration options for qpid	281
4.27. Description of configuration options for zeromq	282
4.28. Description of configuration options for amqp	282
4.29. Description of configuration options for rpc	282
5.1. Description of configuration options for api	284
5.2. Description of configuration options for assignment	285
5.3. Description of configuration options for auth	286
5.4. Description of configuration options for cache	286
5.5. Description of configuration options for catalog	286
5.6. Description of configuration options for credential	287
5.7. Description of configuration options for database	287
5.8. Description of configuration options for debug	287
5.9. Description of configuration options for ec2	288
5.10. Description of configuration options for federation	288
5.11. Description of configuration options for identity	288
5.12. Description of configuration options for kvs	289
5.13. Description of configuration options for ldap	289
5.14. Description of configuration options for logging	292
5.15. Description of configuration options for memcache	293
5.16. Description of configuration options for misc	293
5.17. Description of configuration options for notification	293
5.18. Description of configuration options for oauth	293
5.19. Description of configuration options for os_inherit	293
5.20. Description of configuration options for policy	293
5.21. Description of configuration options for revoke	294
5.22. Description of configuration options for security	294
5.23. Description of configuration options for ssl	294
5.24. Description of configuration options for stats	295
5.25. Description of configuration options for token	295
5.26. Description of configuration options for trust	295
5.27. Description of configuration options for rpc	295
5.28. Description of configuration options for amqp	296
5.29. Description of configuration options for qpid	296
5.30. Description of configuration options for rabbit	296
5.31. Description of configuration options for zeromq	297
5.32. Description of configuration options for redis	297
5.33. New options	326
5.34. New default values	328
5.35. Deprecated options	329
6.1. Description of configuration options for auth_token	330
6.2. Description of configuration options for common	331
6.3. Description of configuration options for db	332
6.4. Description of configuration options for imagecache	333
6.5. Description of configuration options for logging	334

6.6. Description of configuration options for matchmaker	335
6.7. Description of configuration options for paste	335
6.8. Description of configuration options for policy	335
6.9. Description of configuration options for redis	335
6.10. Description of configuration options for registry	335
6.11. Description of configuration options for testing	336
6.12. Description of configuration options for api	336
6.13. Description of configuration options for ssl	337
6.14. Description of configuration options for wsgi	337
6.15. Description of configuration options for rabbitmq	338
6.16. Description of configuration options for qpid	339
6.17. Description of configuration options for zmq	339
6.18. Description of configuration options for amqp	340
6.19. Description of configuration options for rpc	340
6.20. Description of configuration options for cinder	341
6.21. Description of configuration options for filesystem	341
6.22. Description of configuration options for gridfs	341
6.23. Description of configuration options for rbd	342
6.24. Description of configuration options for s3	342
6.25. Description of configuration options for sheepdog	342
6.26. Description of configuration options for swift	342
6.27. Description of configuration options for vmware	344
6.28. New options	365
6.29. New default values	367
6.30. Deprecated options	367
7.1. Description of configuration options for common	368
7.2. Description of configuration options for bigswitch	370
7.3. Description of configuration options for brocade	371
7.4. Description of configuration options for cisco	371
7.5. Description of configuration options for hyperv_agent	372
7.6. Description of configuration options for embrane	373
7.7. Description of configuration options for sdnve	373
7.8. Description of configuration options for linuxbridge_agent	374
7.9. Description of configuration options for mlnx	374
7.10. Description of configuration options for meta	375
7.11. Description of configuration options for ml2	376
7.12. Description of configuration options for ml2_flat	376
7.13. Description of configuration options for ml2_gre	376
7.14. Description of configuration options for ml2_vlan	376
7.15. Description of configuration options for ml2_vxlan	376
7.16. Description of configuration options for ml2_arista	377
7.17. Description of configuration options for ml2_bigswitch	377
7.18. Description of configuration options for ml2_brocade	379
7.19. Description of configuration options for ml2_cisco	379
7.20. Description of configuration options for ml2_mlnx	379
7.21. Description of configuration options for ml2_odl	379
7.22. Description of configuration options for ml2_ofa	379
7.23. Description of configuration options for ml2_l2pop	380
7.24. Description of configuration options for ml2_ncs	380
7.25. Description of configuration options for midonet	380
7.26. Description of configuration options for nec	380

7.27. Description of configuration options for nuage	381
7.28. Description of configuration options for nvsd	381
7.29. Description of configuration options for vmware	382
7.30. Description of configuration options for openswitch_agent	383
7.31. Description of configuration options for plumgrid	384
7.32. Description of configuration options for ryu	385
7.33. Description of configuration options for rabbitmq	385
7.34. Description of configuration options for kombu	386
7.35. Description of configuration options for qpid	387
7.36. Description of configuration options for zeromq	387
7.37. Description of configuration options for rpc	388
7.38. Description of configuration options for notifier	388
7.39. Description of configuration options for redis	389
7.40. Description of configuration options for agent	389
7.41. Description of configuration options for api	389
7.42. Description of configuration options for auth_token	390
7.43. Description of configuration options for cadf	391
7.44. Description of configuration options for compute	391
7.45. Description of configuration options for db	392
7.46. Description of configuration options for dhcp_agent	393
7.47. Description of configuration options for embrane_lb	393
7.48. Description of configuration options for fwaas	393
7.49. Description of configuration options for l3_agent	394
7.50. Description of configuration options for lbaas	394
7.51. Description of configuration options for logging	395
7.52. Description of configuration options for metadata	396
7.53. Description of configuration options for metering_agent	396
7.54. Description of configuration options for policy	397
7.55. Description of configuration options for quotas	397
7.56. Description of configuration options for rootwrap	398
7.57. Description of configuration options for scheduler	398
7.58. Description of configuration options for securitygroups	399
7.59. Description of configuration options for ssl	399
7.60. Description of configuration options for testing	399
7.61. Description of configuration options for varmour	400
7.62. Description of configuration options for vpn	400
7.63. Description of configuration options for wsgi	400
7.64. Log files used by Networking services	401
7.65. New options	418
7.66. New default values	424
7.67. Deprecated options	424
8.1. Description of configuration options for [swift-hash] in swift.conf-sample	427
8.2. Description of configuration options for [DEFAULT] in object-server.conf-sample	427
8.3. Description of configuration options for [app:object-server] in object-server.conf-sample	429
8.4. Description of configuration options for [pipeline:main] in object-server.conf-sample	429
8.5. Description of configuration options for [object-replicator] in object-server.conf-sample	430

8.6. Description of configuration options for [object-updater] in object-server.conf-sample	430
8.7. Description of configuration options for [object-auditor] in object-server.conf-sample	431
8.8. Description of configuration options for [filter:healthcheck] in object-server.conf-sample	431
8.9. Description of configuration options for [filter:recon] in object-server.conf-sample	431
8.10. Description of configuration options for [DEFAULT] in object-expirer.conf-sample	436
8.11. Description of configuration options for [app:proxy-server] in object-expirer.conf-sample	436
8.12. Description of configuration options for [filter:cache] in object-expirer.conf-sample	436
8.13. Description of configuration options for [filter:catch_errors] in object-expirer.conf-sample	436
8.14. Description of configuration options for [object-expirer] in object-expirer.conf-sample	436
8.15. Description of configuration options for [pipeline:main] in object-expirer.conf-sample	437
8.16. Description of configuration options for [DEFAULT] in container-server.conf-sample	438
8.17. Description of configuration options for [app:container-server] in container-server.conf-sample	439
8.18. Description of configuration options for [pipeline:main] in container-server.conf-sample	440
8.19. Description of configuration options for [container-replicator] in container-server.conf-sample	440
8.20. Description of configuration options for [container-updater] in container-server.conf-sample	440
8.21. Description of configuration options for [container-auditor] in container-server.conf-sample	440
8.22. Description of configuration options for [container-sync] in container-server.conf-sample	441
8.23. Description of configuration options for [filter:healthcheck] in container-server.conf-sample	441
8.24. Description of configuration options for [filter:recon] in container-server.conf-sample	441
8.25. Description of configuration options for [DEFAULT] in account-server.conf-sample	444
8.26. Description of configuration options for [app:account-server] in account-server.conf-sample	445
8.27. Description of configuration options for [pipeline:main] in account-server.conf-sample	446
8.28. Description of configuration options for [account-replicator] in account-server.conf-sample	446
8.29. Description of configuration options for [account-auditor] in account-server.conf-sample	446
8.30. Description of configuration options for [account-reaper] in account-server.conf-sample	447

8.57. Description of configuration options for [filter:name_check] in proxy-server.conf-sample	473
8.58. Description of configuration options for [swift-constraints] in swift.conf-sample	473
8.59. Description of configuration options for [dispersion] in dispersion.conf-sample	476
8.60. Description of configuration options for [filter:slo] in proxy-server.conf-sample	476
8.61. Description of configuration options for [filter:container-quotas] in proxy-server.conf-sample	477
8.62. Description of configuration options for [filter:bulk] in proxy-server.conf-sample	478
8.63. Description of configuration options for [drive-audit] in drive-audit.conf-sample	480
8.64. Description of configuration options for [filter:formpost] in proxy-server.conf-sample	481
8.65. Description of configuration options for [filter:staticweb] in proxy-server.conf-sample	482
9.1. Description of configuration options for auth_token	483
9.2. Description of configuration options for common	484
9.3. Description of configuration options for crypt	485
9.4. Description of configuration options for database	485
9.5. Description of configuration options for debug	486
9.6. Description of configuration options for loadbalancer	486
9.7. Description of configuration options for logging	486
9.8. Description of configuration options for quota	487
9.9. Description of configuration options for redis	487
9.10. Description of configuration options for api	487
9.11. Description of configuration options for cfn_api	489
9.12. Description of configuration options for cloudwatch_api	489
9.13. Description of configuration options for metadata_api	490
9.14. Description of configuration options for waitcondition_api	490
9.15. Description of configuration options for clients	490
9.16. Description of configuration options for clients_backends	490
9.17. Description of configuration options for clients_ceilometer	491
9.18. Description of configuration options for clients_cinder	491
9.19. Description of configuration options for clients_heat	491
9.20. Description of configuration options for clients_keystone	491
9.21. Description of configuration options for clients_neutron	492
9.22. Description of configuration options for clients_nova	492
9.23. Description of configuration options for clients_swift	492
9.24. Description of configuration options for clients_trove	492
9.25. Description of configuration options for rabbitmq	493
9.26. Description of configuration options for qpid	494
9.27. Description of configuration options for zeromq	495
9.28. Description of configuration options for amqp	495
9.29. Description of configuration options for rpc	495
9.30. Description of configuration options for notification	496
9.31. New options	496
9.32. New default values	498
9.33. Deprecated options	499

OpenStack configuration overview

OpenStack is a collection of open source project components that enable setting up cloud services. Each component uses similar configuration techniques and a common framework for INI file options.

This guide pulls together multiple references and configuration options for the following OpenStack components:

- OpenStack Block Storage
- OpenStack Compute
- OpenStack Dashboard
- Database Service
- OpenStack Identity
- OpenStack Image Service
- OpenStack Networking
- OpenStack Object Storage
- Telemetry
- Orchestration

Conventions

The OpenStack documentation uses several typesetting conventions.

Notices

Notices take three forms:



Note

The information in a note is usually in the form of a handy tip or reminder.



Important

The information in an important notice is something you must be aware of before proceeding.



Warning

The information in warnings is critical. Warnings provide additional information about risk of data loss or security issues.

Command prompts

Commands prefixed with the # prompt are to be executed by the `root` user. These examples can also be executed by using the `sudo` command, if available.

Commands prefixed with the \$ prompt can be executed by any user, including `root`.

Document change history

This version of the guide replaces and obsoletes all previous versions. The following table describes the most recent changes:

Revision Date	Summary of Changes
April 16, 2014	<ul style="list-style-type: none">Update for Icehouse: Updated all configuration tables, include sample configuration files, add chapters for Database Service, Orchestration, and Telemetry.
March 11, 2014	<ul style="list-style-type: none">Sorted component listing. Moved procedures to the Cloud Administrator Guide
January 9, 2014	<ul style="list-style-type: none">Removes content addressed in installation, merges duplicated content, and revises legacy references.
October 17, 2013	<ul style="list-style-type: none">Havana release.
August 16, 2013	<ul style="list-style-type: none">Moves Block Storage driver configuration information from the <i>Block Storage Administration Guide</i> to this reference.
June 10, 2013	<ul style="list-style-type: none">Initial creation of Configuration Reference.

Configuration file format

OpenStack uses the *INI* file format for configuration files. An INI file is a simple text file that specifies options as `key=value` pairs, grouped into sections. The `DEFAULT` section contains most of the configuration options. Lines starting with a hash sign (#) are comment lines. For example:

```
[DEFAULT]
# Print debugging output (set logging level to DEBUG instead
# of default WARNING level). (boolean value)
debug = true
# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose = true

[database]
# The SQLAlchemy connection string used to connect to the
# database (string value)
connection = mysql://keystone:KEYSTONE_DBPASS@controller/keystone
```

Options can have different type for values. The comments in the sample config files always mention these. The following types are used by OpenStack:

boolean value Enables or disables an option. The allowed values are `true` and `false`.

```
# Enable the experimental use of database reconnect
on
# connection lost (boolean value)
use_db_reconnect = false
```

floating point value	<p>A floating point number like 0.25 or 1000.</p> <pre># Sleep time in seconds for polling an ongoing async task # (floating point value) task_poll_interval = 0.5</pre>
integer value	<p>An integer number is a number without fractional components, like 0 or 42.</p> <pre># The port which the OpenStack Compute service listens on. # (integer value) compute_port = 8774</pre>
list value	<p>Represents values of other types, separated by commas. As an example, the following sets <code>allowed_rpc_exception_modules</code> to a list containing the four elements <code>oslo.messaging.exceptions</code>, <code>nova.exception</code>, <code>cinder.exception</code>, and <code>exceptions</code>:</p> <pre># Modules of exceptions that are permitted to be recreated # upon receiving exception data from an rpc call. (list value) allowed_rpc_exception_modules = oslo.messaging. exceptions,nova.exception,cinder.exception, exceptions</pre>
multi valued	<p>A multi-valued option is a string value and can be given more than once, all values will be used.</p> <pre># Driver or drivers to handle sending notifications. (multi # valued) notification_driver = nova.openstack.common. notifier.rpc_notifier notification_driver = ceilometer.compute. nova_notifier</pre>
string value	<p>Strings can be optionally enclosed with single or double quotes.</p> <pre># onready allows you to send a notification when the process # is ready to serve. For example, to have it notify using # systemd, one could set shell command: "onready = systemd- # notify --ready" or a module with notify() method: "onready = # keystone.common.systemd". (string value) onready = systemd-notify --ready # If an instance is passed with the log message, format it # like this (string value) instance_format = "[instance: %(uuid)s] "</pre>

Sections

Configuration options are grouped by section. Most configuration file supports at least the following sections:

- [DEFAULT] Contains most configuration options. If the documentation for a configuration option does not specify its section, assume that it appears in this section.
- [database] Configuration options for the database that stores the state of the OpenStack service.

Substitution

The configuration file supports variable substitution. After you set a configuration option, it can be referenced in later configuration values when you precede it with a `$`, like `$OPTION`.

The following example uses the values of `rabbit_host` and `rabbit_port` to define the value of the `rabbit_hosts` option, in this case as `controller:5672`.

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller

# The RabbitMQ broker port where a single node is used.
# (integer value)
rabbit_port = 5672

# RabbitMQ HA cluster host:port pairs. (list value)
rabbit_hosts = $rabbit_host:$rabbit_port
```

To avoid substitution, use `$$`, it is replaced by a single `$`. For example, if your LDAP DNS password is `$xkj432`, specify it, as follows:

```
ldap_dns_password = $$xkj432
```

The code uses the Python `string.Template.safe_substitute()` method to implement variable substitution. For more details on how variable substitution is resolved, see <http://docs.python.org/2/library/string.html#template-strings> and [PEP 292](#).

Whitespace

To include whitespace in a configuration value, use a quoted string. For example:

```
ldap_dns_password='a password with spaces'
```

Define an alternate location for a config file

Most services and the `*-manage` command-line clients load the configuration file. To define an alternate location for the configuration file, pass the `--config-file CONFIG_FILE` parameter when you start a service or call a `*-manage` command.

1. Block Storage

Table of Contents

Introduction to the Block Storage service	1
<code>cinder.conf</code> configuration file	2
Volume drivers	3
Backup drivers	77
Block Storage sample configuration files	79
Log files used by Block Storage	121
Fibre Channel Zone Manager	121
Additional options	123
New, updated and deprecated options for <code>cinder</code>	136

The OpenStack Block Storage service works with many different storage drivers that you can configure by using these instructions.

Introduction to the Block Storage service

The OpenStack Block Storage service provides persistent block storage resources that OpenStack Compute instances can consume. This includes secondary attached storage similar to the Amazon Elastic Block Storage (EBS) offering. In addition, you can write images to a Block Storage device for Compute to use as a bootable persistent instance.

The Block Storage service differs slightly from the Amazon EBS offering. The Block Storage service does not provide a shared storage solution like NFS. With the Block Storage service, you can attach a device to only one instance.

The Block Storage service provides:

- `cinder-api`. A WSGI app that authenticates and routes requests throughout the Block Storage service. It supports the OpenStack APIs only, although there is a translation that can be done through Compute's EC2 interface, which calls in to the Block Storage client.
- `cinder-scheduler`. Schedules and routes requests to the appropriate volume service. Depending upon your configuration, this may be simple round-robin scheduling to the running volume services, or it can be more sophisticated through the use of the Filter Scheduler. The Filter Scheduler is the default and enables filters on things like Capacity, Availability Zone, Volume Types, and Capabilities as well as custom filters.
- `cinder-volume`. Manages Block Storage devices, specifically the back-end devices themselves.
- `cinder-backup`. Provides a means to back up a Block Storage Volume to OpenStack Object Store (SWIFT).

The Block Storage service contains the following components:

- **Back-end Storage Devices**. The Block Storage service requires some form of back-end storage that the service is built on. The default implementation is to use LVM on a local

volume group named "cinder-volumes." In addition to the base driver implementation, the Block Storage service also provides the means to add support for other storage devices to be utilized such as external Raid Arrays or other storage appliances. These back-end storage devices may have custom block sizes when using KVM or QEMU as the hypervisor.

- **Users and Tenants (Projects).** The Block Storage service can be used by many different cloud computing consumers or customers (tenants on a shared system), using role-based access assignments. Roles control the actions that a user is allowed to perform. In the default configuration, most actions do not require a particular role, but this can be configured by the system administrator in the appropriate `policy.json` file that maintains the rules. A user's access to particular volumes is limited by tenant, but the username and password are assigned per user. Key pairs granting access to a volume are enabled per user, but quotas to control resource consumption across available hardware resources are per tenant.

For tenants, quota controls are available to limit:

- The number of volumes that can be created.
- The number of snapshots that can be created.
- The total number of GBs allowed per tenant (shared between snapshots and volumes).

You can revise the default quota values with the Block Storage CLI, so the limits placed by quotas are editable by admin users.

- **Volumes, Snapshots, and Backups.** The basic resources offered by the Block Storage service are volumes and snapshots which are derived from volumes and volume backups:
 - **Volumes.** Allocated block storage resources that can be attached to instances as secondary storage or they can be used as the root store to boot instances. Volumes are persistent R/W block storage devices most commonly attached to the compute node through iSCSI.
 - **Snapshots.** A read-only point in time copy of a volume. The snapshot can be created from a volume that is currently in use (through the use of `--force True`) or in an available state. The snapshot can then be used to create a new volume through `create from snapshot`.
 - **Backups.** An archived copy of a volume currently stored in OpenStack Object Storage (Swift).

cinder.conf configuration file

The `cinder.conf` file is installed in `/etc/cinder` by default. When you manually install the Block Storage service, the options in the `cinder.conf` file are set to default values.

This example shows a typical `cinder.conf` file:

```
[DEFAULT]
rootwrap_config=/etc/cinder/rootwrap.conf
sql_connection = mysql://cinder:openstack@192.168.127.130/cinder
```

```
api_paste_config = /etc/cinder/api-paste.ini

iscsi_helper=tgtadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
#osapi_volume_listen_port=5900

# Add these when not using the defaults.
rabbit_host = 10.10.10.10
rabbit_port = 5672
rabbit_userid = rabbit
rabbit_password = secure_password
rabbit_virtual_host = /nova
```

Volume drivers

To use different volume drivers for the `cinder-volume` service, use the parameters described in these sections.

The volume drivers are included in the Block Storage repository (<https://github.com/openstack/cinder>). To set a volume driver, use the `volume_driver` flag. The default is:

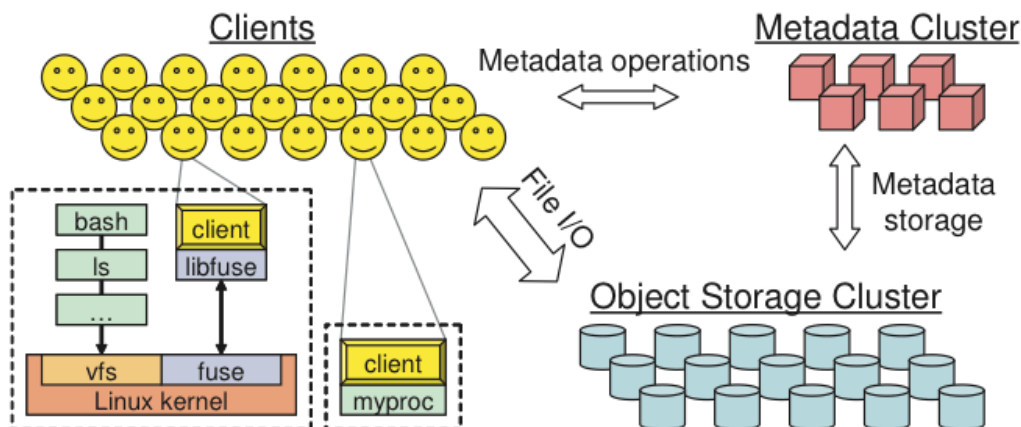
```
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
```

Ceph RADOS Block Device (RBD)

If you use KVM or QEMU as your hypervisor, you can configure the Compute service to use [Ceph RADOS block devices \(RBD\)](#) for volumes.

Ceph is a massively scalable, open source, distributed storage system. It is comprised of an object store, block store, and a POSIX-compliant distributed file system. The platform can auto-scale to the exabyte level and beyond. It runs on commodity hardware, is self-healing and self-managing, and has no single point of failure. Ceph is in the Linux kernel and is integrated with the OpenStack cloud operating system. Due to its open-source nature, you can install and use this portable storage platform in public or private clouds.

Figure 1.1. Ceph architecture



RADOS

Ceph is based on *RADOS: Reliable Autonomic Distributed Object Store*. RADOS distributes objects across the storage cluster and replicates objects for fault tolerance. RADOS contains the following major components:

- *Object Storage Device (OSD) Daemon*. The storage daemon for the RADOS service, which interacts with the OSD (physical or logical storage unit for your data).

You must run this daemon on each server in your cluster. For each OSD, you can have an associated hard drive disk. For performance purposes, pool your hard drive disk with raid arrays, logical volume management (LVM), or B-tree file system (`Btrfs`) pooling. By default, the following pools are created: data, metadata, and RBD.

- *Meta-Data Server (MDS)*. Stores metadata. MDSs build a POSIX file system on top of objects for Ceph clients. However, if you do not use the Ceph file system, you do not need a metadata server.
- *Monitor (MON)*. A lightweight daemon that handles all communications with external applications and clients. It also provides a consensus for distributed decision making in a Ceph/RADOS cluster. For instance, when you mount a Ceph shared on a client, you point to the address of a MON server. It checks the state and the consistency of the data. In an ideal setup, you must run at least three `ceph-mon` daemons on separate servers.

Ceph developers recommend that you use `Btrfs` as a file system for storage. XFS might be a better alternative for production environments; XFS is an excellent alternative to `Btrfs`. The `ext4` file system is also compatible but does not exploit the power of Ceph.



Note

If using `Btrfs`, ensure that you use the correct version (see [Ceph Dependencies](#)).

For more information about usable file systems, see ceph.com/ceph-storage/file-system/.

Ways to store, use, and expose data

To store and access your data, you can use the following storage systems:

- *RADOS*. Use as an object, default storage mechanism.
- *RBD*. Use as a block device. The Linux kernel RBD (rados block device) driver allows striping a Linux block device over multiple distributed object store data objects. It is compatible with the KVM RBD image.
- *CephFS*. Use as a file, POSIX-compliant file system.

Ceph exposes RADOS; you can access it through the following interfaces:

- *RADOS Gateway*. OpenStack Object Storage and Amazon-S3 compatible RESTful interface (see [RADOS_Gateway](#)).

- *librados*, and its related C/C++ bindings.
- *rbd* and *QEMU-RBD*. Linux kernel and QEMU block devices that stripe data across multiple objects.

Driver options

The following table contains the configuration options supported by the Ceph RADOS Block Device driver.

Table 1.1. Description of configuration options for storage_ceph

Configuration option = Default value	Description
[DEFAULT]	
rbd_ceph_conf =	(StrOpt) path to the ceph configuration file to use
rbd_flatten_volume_from_snapshot = False	(BoolOpt) flatten volumes created from snapshots to remove dependency
rbd_max_clone_depth = 5	(IntOpt) maximum number of nested clones that can be taken of a volume before enforcing a flatten prior to next clone. A value of zero disables cloning
rbd_pool = rbd	(StrOpt) the RADOS pool in which rbd volumes are stored
rbd_secret_uuid = None	(StrOpt) the libvirt uuid of the secret for the rbd_uservolumes
rbd_user = None	(StrOpt) the RADOS client name for accessing rbd volumes - only set when using cephx authentication
volume_tmp_dir = None	(StrOpt) where to store temporary image files if the volume driver does not write them directly to the volume

Coraid AoE driver configuration

Coraid storage appliances can provide block-level storage to OpenStack instances. Coraid storage appliances use the low-latency ATA-over-Ethernet (ATA) protocol to provide high-bandwidth data transfer between hosts and data on the network.

Once configured for OpenStack, you can:

- Create, delete, attach, and detach block storage volumes.
- Create, list, and delete volume snapshots.
- Create a volume from a snapshot, copy an image to a volume, copy a volume to an image, clone a volume, and get volume statistics.

This document describes how to configure the OpenStack Block Storage service for use with Coraid storage appliances.

Terminology

These terms are used in this section:

Term	Definition
AoE	ATA-over-Ethernet protocol

Term	Definition
EtherCloud Storage Manager (ESM)	ESM provides live monitoring and management of EtherDrive appliances that use the AoE protocol, such as the SRX and VSX.
Fully-Qualified Repository Name (FQRN)	The FQRN is the full identifier of a storage profile. FQRN syntax is: <i>performance_class-availability_class:profile_name:repository_name</i>
SAN	Storage Area Network
SRX	Coraid EtherDrive SRX block storage appliance
VSX	Coraid EtherDrive VSX storage virtualization appliance

Requirements

To support the OpenStack Block Storage service, your SAN must include an SRX for physical storage, a VSX running at least CorOS v2.0.6 for snapshot support, and an ESM running at least v2.1.1 for storage repository orchestration. Ensure that all storage appliances are installed and connected to your network before you configure OpenStack volumes.

In order for the node to communicate with the SAN, you must install the Coraid AoE Linux driver on each Compute node on the network that runs an OpenStack instance.

Overview

To configure the OpenStack Block Storage for use with Coraid storage appliances, perform the following procedures:

1. [Download and install the Coraid Linux AoE driver.](#)
2. [Create a storage profile by using the Coraid ESM GUI.](#)
3. [Create a storage repository by using the ESM GUI and record the FQRN.](#)
4. [Configure the `cinder.conf` file.](#)
5. [Create and associate a block storage volume type.](#)

Install the Coraid AoE driver

Install the Coraid AoE driver on every compute node that will require access to block storage.

The latest AoE drivers will always be located at <http://support.coraid.com/support/linux/>.

To download and install the AoE driver, follow the instructions below, replacing "aoeXXX" with the AoE driver file name:

1. Download the latest Coraid AoE driver.

```
$ wget http://support.coraid.com/support/linux/aoeXXX.tar.gz
```

2. Unpack the AoE driver.

3. Install the AoE driver.

```
$ cd aoeXXX
```

```
$ make
```

```
# make install
```

4. Initialize the AoE driver.

```
# modprobe aoe
```

5. Optionally, specify the Ethernet interfaces that the node can use to communicate with the SAN.

The AoE driver may use every Ethernet interface available to the node unless limited with the `aoe_iflist` parameter. For more information about the `aoe_iflist` parameter, see the `aoe_readme` file included with the AoE driver.

```
# modprobe aoe_iflist="eth1 eth2 ..."
```

Create a storage profile

To create a storage profile using the ESM GUI:

1. Log in to the ESM.
2. Click **Storage Profiles** in the **SAN Domain** pane.
3. Choose **Menu > Create Storage Profile**. If the option is unavailable, you might not have appropriate permissions. Make sure you are logged in to the ESM as the SAN administrator.
4. Use the storage class selector to select a storage class.

Each storage class includes performance and availability criteria (see the Storage Classes topic in the ESM Online Help for information on the different options).

5. Select a RAID type (if more than one is available) for the selected profile type.
6. Type a **Storage Profile** name.

The name is restricted to alphanumeric characters, underscore (`_`), and hyphen (`-`), and cannot exceed 32 characters.
7. Select the drive size from the drop-down menu.
8. Select the number of drives to be initialized for each RAID (LUN) from the drop-down menu (if the selected RAID type requires multiple drives).
9. Type the number of RAID sets (LUNs) you want to create in the repository by using this profile.
10. Click **Next**.

Create a storage repository and get the FQRN

Create a storage repository and get its fully qualified repository name (FQRN):

1. Access the **Create Storage Repository** dialog box.
2. Type a Storage Repository name.

The name is restricted to alphanumeric characters, underscore (_), hyphen (-), and cannot exceed 32 characters.

3. Click **Limited** or **Unlimited** to indicate the maximum repository size.

Limited sets the amount of space that can be allocated to the repository. Specify the size in TB, GB, or MB.

When the difference between the reserved space and the space already allocated to LUNs is less than is required by a LUN allocation request, the reserved space is increased until the repository limit is reached.



Note

The reserved space does not include space used for parity or space used for mirrors. If parity and/or mirrors are required, the actual space allocated to the repository from the SAN is greater than that specified in reserved space.

Unlimited—Unlimited means that the amount of space allocated to the repository is unlimited and additional space is allocated to the repository automatically when space is required and available.



Note

Drives specified in the associated Storage Profile must be available on the SAN in order to allocate additional resources.

4. Check the **Resizeable LUN** box.

This is required for OpenStack volumes.



Note

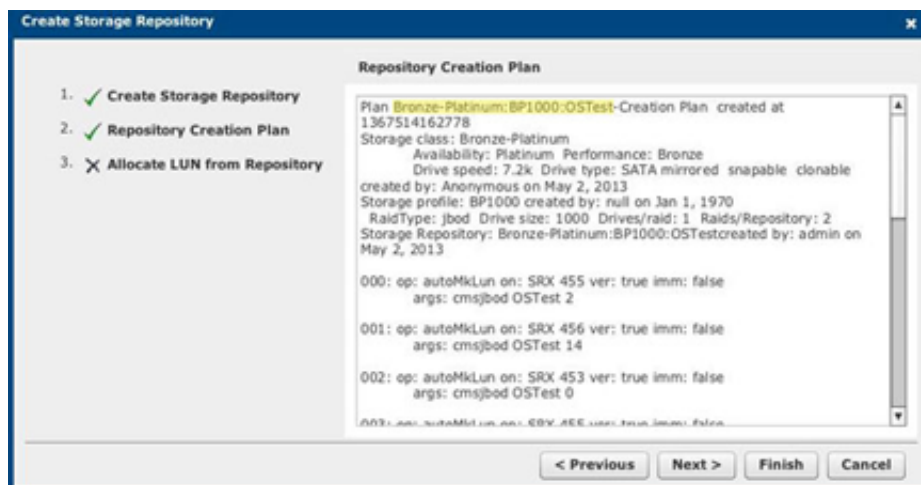
If the Storage Profile associated with the repository has platinum availability, the **Resizeable LUN** box is automatically checked.

5. Check the **Show Allocation Plan API calls** box. Click **Next**.
6. Record the FQRN and click **Finish**.

The FQRN is located in the first line of output following the `Plan` keyword in the **Repository Creation Plan** window. The FQRN syntax is `performance_class-availability_class:profile_name:repository_name`.

In this example, the FQRN is `Bronze-Platinum:BP1000:OSTest`, and is highlighted.

Figure 1.2. Repository Creation Plan screen



Record the FQRN; it is a required parameter later in the configuration procedure.

Configure options in the `cinder.conf` file

Edit or add the following lines to the file `/etc/cinder/cinder.conf`:

```
volume_driver = cinder.volume.drivers.coraid.CoraidDriver
coraid_esm_address = ESM_IP_address
coraid_user = username
coraid_group = Access_Control_Group_name
coraid_password = password
coraid_repository_key = coraid_repository_key
```

Table 1.2. Description of configuration options for coraid

Configuration option = Default value	Description
[DEFAULT]	
<code>coraid_esm_address =</code>	(StrOpt) IP address of Coraid ESM
<code>coraid_group = admin</code>	(StrOpt) Name of group on Coraid ESM to which <code>coraid_user</code> belongs (must have admin privilege)
<code>coraid_password = password</code>	(StrOpt) Password to connect to Coraid ESM
<code>coraid_repository_key = coraid_repository</code>	(StrOpt) Volume Type key name to store ESM Repository Name
<code>coraid_user = admin</code>	(StrOpt) User name to connect to Coraid ESM

Access to storage devices and storage repositories can be controlled using Access Control Groups configured in ESM. Configuring `cinder.conf` to log on to ESM as the SAN administrator (user name `admin`), will grant full access to the devices and repositories configured in ESM.

Optionally, you can configure an ESM Access Control Group and user. Then, use the `cinder.conf` file to configure access to the ESM through that group, and user limits

access from the OpenStack instance to devices and storage repositories that are defined in the group.

To manage access to the SAN by using Access Control Groups, you must enable the Use Access Control setting in the **ESM System Setup > Security** screen.

For more information, see the ESM Online Help.

Create and associate a volume type

Create and associate a volume with the ESM storage repository.

1. Restart Cinder.

```
# service openstack-cinder-api restart
# service openstack-cinder-scheduler restart
# service openstack-cinder-volume restart
```

2. Create a volume.

```
$ cinder type-create 'volume_type_name'
```

where *volume_type_name* is the name you assign the volume. You will see output similar to the following:

```
+-----+-----+
|              ID              |      Name      |
+-----+-----+
| 7fa6b5ab-3e20-40f0-b773-dd9e16778722 | JBOD-SAS600 |
+-----+-----+
```

Record the value in the ID field; you use this value in the next step.

3. Associate the volume type with the Storage Repository.

```
# cinder type-key UUID set coraid_repository_key='FQRN'
```

Variable	Description
<i>UUID</i>	The ID returned from the <code>cinder type-create</code> command. You can use the <code>cinder type-list</code> command to recover the ID.
<i>coraid_repository_key</i>	The key name used to associate the Cinder volume type with the ESM in the <code>cinder.conf</code> file. If no key name was defined, this is default value for <code>coraid_repository</code> .
<i>FQRN</i>	The FQRN recorded during the Create Storage Repository process.

Dell EqualLogic volume driver

The Dell EqualLogic volume driver interacts with configured EqualLogic arrays and supports various operations, such as volume creation and deletion, volume attachment and detachment, snapshot creation and deletion, and clone creation.

To configure and use a Dell EqualLogic array with Block Storage, modify your `cinder.conf` as follows.

Set the `volume_driver` option to the Dell EqualLogic volume driver:

```
volume_driver=cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
```

Set the `san_ip` option to the IP address to reach the EqualLogic Group through SSH:

```
san_ip=10.10.72.53
```

Set the `san_login` option to the user name to login to the Group manager:

```
san_login=grpadmin
```

Set the `san_password` option to the password to login the Group manager with:

```
san_password=password
```

The following table describes additional options that the driver supports:

Table 1.3. Description of configuration options for eqlx

Configuration option = Default value	Description
[DEFAULT]	
<code>eqlx_chap_login = admin</code>	(StrOpt) Existing CHAP account name
<code>eqlx_chap_password = password</code>	(StrOpt) Password for specified CHAP account name
<code>eqlx_cli_max_retries = 5</code>	(IntOpt) Maximum retry count for reconnection
<code>eqlx_cli_timeout = 30</code>	(IntOpt) Timeout for the Group Manager cli command execution
<code>eqlx_group_name = group-0</code>	(StrOpt) Group name to use for creating volumes
<code>eqlx_pool = default</code>	(StrOpt) Pool in which volumes will be created
<code>eqlx_use_chap = False</code>	(BoolOpt) Use CHAP authentication for targets?

EMC VNX direct driver

Use the EMC VNX direct driver to create, attach, detach, and delete volumes, create and delete snapshots, and so on. This driver is based on the Cinder-defined `ISCSIDriver` driver.

To complete volume operations, the driver uses the NaviSec command-line interface (CLI) to communicate with back-end EMC VNX storage.

System requirements

- Flare version 5.32 or later.
- You must activate VNX Snapshot and Thin Provisioning license for the array. Ensure that all the iSCSI ports from the VNX are accessible through OpenStack hosts.
- Navisphere CLI v7.32 or later.

EMC storage VNX Series are supported.

Supported operations

- Create volume

- Delete volume
- Attach volume
- Detach volume
- Create snapshot
- Delete snapshot
- Create volume from snapshot
- Create cloned volume
- Copy image to volume
- Copy volume to image
- Extend volume

Set up the VNX direct driver

Complete these high-level tasks to set up the VNX direct driver:

1. Install NaviSecCLI. You must install the NaviSecCLI tool on the controller node and all the Cinder nodes in an OpenStack deployment. See [the section called “Install NaviSecCLI” \[12\]](#).
2. Register with VNX. See [the section called “Register with VNX” \[12\]](#)

Install NaviSecCLI

On Ubuntu x64, download the NaviSecCLI deb package from [EMC's OpenStack Github](#) web site.

For all the other variants of Linux, download the NaviSecCLI rpm package from EMC's support web site for [VNX2 series](#) or [VNX1 series](#). Login is required.

Register with VNX

To export a VNX volume to a compute node or a volume node, you must register the node with VNX.

Procedure 1.1. To register the node

1. On the compute node or volume node 1.1.1.1, do the following (assume 10.10.61.35 is the iSCSI target):

```
# /etc/init.d/open-iscsi start
# iscsiadm -m discovery -t st -p 10.10.61.35
# cd /etc/iscsi
# more initiatorname.iscsi
# iscsiadm -m node
```

2. Log in to VNX from the node using the target corresponding to the SPA port:

```
# iscsiadm -m node -T iqn.1992-04.com.emc:cx.apm01234567890.a0 -p 10.10.61.35 -l
```

Where `iqn.1992-04.com.emc:cx.apm01234567890.a0` is the initiator name of the node. Login to Unisphere, go to `VNX00000->Hosts->Initiators`, Refresh and wait until initiator `iqn.1992-04.com.emc:cx.apm01234567890.a0` with SP Port `A-8v0` appears.

3. Click **Register**, select **CLARiiON/VNX**, and enter the host name `myhost1` and IP address `myhost1`. Click **Register**. Now host `1.1.1.1` also appears under `Hosts->Host List`.

4. Log out of VNX on the node:

```
# iscsiadm -m node -u
```

5. Log in to VNX from the node using the target corresponding to the SPB port:

```
# iscsiadm -m node -T iqn.1992-04.com.emc:cx.apm01234567890.b8 -p 10.10.10.11 -l
```

6. In Unisphere register the initiator with the SPB port.

7. Log out:

```
# iscsiadm -m node -u
```

cinder.conf configuration file

Make the following changes in `/etc/cinder/cinder.conf`.

For the VNX iSCSI driver, add the following entries, where `10.10.61.35` is the IP address of the VNX iSCSI target, `10.10.72.41` is the IP address of the VNX array (SPA or SPB), `default_timeout` is the default time out for CLI operations in minutes, and `max_luns_per_storage_group` is the default max number of LUNs in a storage group:

```
iscsi_ip_address = 10.10.61.35
san_ip = 10.10.72.41
san_login = global_username
san_password = password
naviseccli_path = /opt/Navisphere/bin/naviseccli
storage_vnx_pool_name = poolname
default_timeout = 10
max_luns_per_storage_group=256
volume_driver=cinder.volume.drivers.emc.emc_cli_iscsi.EMCCLIISCSIDriver
```



Note

To find out `max_luns_per_storage_group` for each VNX model, refer to the [EMC's support](#) web site (login is required).

Restart the `cinder-volume` service.

Volume type support

Volume type support allows user to choose thick/thin provisioning capabilities.

Here is an example of how to setup volume type. First create volume types. Then define extra specs for each volume type.

Procedure 1.2. To set up volume types

1. Setup volume types:

```
$ cinder type-create "TypeA"  
$ cinder type-create "TypeB"
```

2. Setup volume type extra specs:

```
$ cinder type-key "TypeA" set storagetype:provisioning=thick  
$ cinder type-key "TypeB" set storagetype:provisioning=thin
```

The previous example creates two volume types: TypeA and TypeB. For TypeA, `storagetype:provisioning` is set to `thick`. Similarly for TypeB, `storagetype:provisioning` is set to `thin`. If `storagetype:provisioning` is not specified, it will be default to `thick`.

EMC SMI-S iSCSI and FC drivers

The EMC volume drivers, `EMCSMISISCSIDriver` and `EMCSMISFCDriver`, has the ability to create/delete and attach/detach volumes and create/delete snapshots, and so on.

The driver runs volume operations by communicating with the backend EMC storage. It uses a CIM client in Python called PyWBEM to perform CIM operations over HTTP.

The EMC CIM Object Manager (ECOM) is packaged with the EMC SMI-S provider. It is a CIM server that enables CIM clients to perform CIM operations over HTTP by using SMI-S in the back-end for EMC storage operations.

The EMC SMI-S Provider supports the SNIA Storage Management Initiative (SMI), an ANSI standard for storage management. It supports VMAX and VNX storage systems.

System requirements

EMC SMI-S Provider V4.6.1 and higher is required. You can download SMI-S from the [EMC's support](#) web site (login is required). See the EMC SMI-S Provider release notes for installation instructions.

EMC storage VMAX Family and VNX Series are supported.

Supported operations

VMAX and VNX arrays support these operations:

- Create volume
- Delete volume
- Attach volume
- Detach volume
- Create snapshot

- Delete snapshot
- Create cloned volume
- Copy image to volume
- Copy volume to image

Only VNX supports the following operations:

- Create volume from snapshot
- Extend volume

Set up the SMI-S drivers

Procedure 1.3. To set up the EMC SMI-S drivers

1. Install the python-pywbem package for your distribution. See [the section called "Install the python-pywbem package" \[15\]](#).
2. Download SMI-S from PowerLink and install it. Add your VNX/VMAX arrays to SMI-S.

For information, see [the section called "Set up SMI-S" \[15\]](#) and the SMI-S release notes.
3. Register with VNX. See [the section called "Register with VNX for the iSCSI driver" \[16\]](#) for the VNX iSCSI driver and [the section called "Register with VNX for the FC driver" \[17\]](#) for the VNX FC driver.
4. Create a masking view on VMAX. See [the section called "Create a masking view on VMAX" \[17\]](#).

Install the python-pywbem package

Install the python-pywbem package for your distribution, as follows:

- On Ubuntu:

```
# apt-get install python-pywbem
```

- On openSUSE:

```
# zypper install python-pywbem
```

- On Fedora:

```
# yum install pywbem
```

Set up SMI-S

You can install SMI-S on a non-OpenStack host. Supported platforms include different flavors of Windows, Red Hat, and SUSE Linux. SMI-S can be installed on a physical server or a VM hosted by an ESX server. Note that the supported hypervisor for a VM running SMI-S is ESX only. See the EMC SMI-S Provider release notes for more information on supported platforms and installation instructions.



Note

You must discover storage arrays on the SMI-S server before you can use the Cinder driver. Follow instructions in the SMI-S release notes.

SMI-S is usually installed at `/opt/emc/ECIM/ECOM/bin` on Linux and `C:\Program Files\EMC\ECIM\ECOM\bin` on Windows. After you install and configure SMI-S, go to that directory and type **TestSmiProvider.exe**.

Use **addsys** in **TestSmiProvider.exe** to add an array. Use **dv** and examine the output after the array is added. Make sure that the arrays are recognized by the SMI-S server before using the EMC Cinder driver.

Register with VNX for the iSCSI driver

To export a VNX volume to a Compute node or a Volume node, you must register the node with VNX.

Procedure 1.4. Register the node

1. On the Compute node or Volume node `1.1.1.1`, do the following (assume `10.10.61.35` is the `iscsi` target):

```
# /etc/init.d/open-iscsi start
# iscsiadm -m discovery -t st -p 10.10.61.35
# cd /etc/iscsi
# more initiatorname.iscsi
# iscsiadm -m node
```

2. Log in to VNX from the node using the target corresponding to the SPA port:

```
# iscsiadm -m node -T iqn.1992-04.com.emc:cx.apm01234567890.a0 -p 10.10.61.35 -l
```

Where `iqn.1992-04.com.emc:cx.apm01234567890.a0` is the initiator name of the node. Log in to Unisphere, go to `VNX00000->Hosts->Initiators`, Refresh and wait until initiator `iqn.1992-04.com.emc:cx.apm01234567890.a0` with SP Port `A-8v0` appears.

3. Click the **Register** button, select **CLARiiON/VNX**, and enter the host name `myhost1` and IP address `myhost1`. Click **Register**. Now host `1.1.1.1` also appears under `Hosts->Host List`.

4. Log out of VNX on the node:

```
# iscsiadm -m node -u
```

5. Log in to VNX from the node using the target corresponding to the SPB port:

```
# iscsiadm -m node -T iqn.1992-04.com.emc:cx.apm01234567890.b8 -p 10.10.10.11 -l
```

6. In Unisphere register the initiator with the SPB port.

7. Log out:

```
# iscsiadm -m node -u
```

Register with VNX for the FC driver

For a VNX volume to be exported to a Compute node or a Volume node, SAN zoning needs to be configured on the node and WWNs of the node need to be registered with VNX in Unisphere.

Create a masking view on VMAX

For VMAX iSCSI and FC drivers, you need to do initial setup in Unisphere for VMAX. In Unisphere for VMAX, create an initiator group, a storage group, and a port group. Put them in a masking view. The initiator group contains the initiator names of the OpenStack hosts. The storage group will contain volumes provisioned by Block Storage.

cinder.conf configuration file

Make the following changes in `/etc/cinder/cinder.conf`.

For VMAX iSCSI driver, add the following entries, where `10.10.61.45` is the IP address of the VMAX iSCSI target:

```
iscsi_target_prefix = iqn.1992-04.com.emc
iscsi_ip_address = 10.10.61.45
volume_driver = cinder.volume.drivers.emc.emc_smis_iscsi.EMCSMISISCSIDriver
cinder_emc_config_file = /etc/cinder/cinder_emc_config.xml
```

For VNX iSCSI driver, add the following entries, where `10.10.61.35` is the IP address of the VNX iSCSI target:

```
iscsi_target_prefix = iqn.2001-07.com.vnx
iscsi_ip_address = 10.10.61.35
volume_driver = cinder.volume.drivers.emc.emc_smis_iscsi.EMCSMISISCSIDriver
cinder_emc_config_file = /etc/cinder/cinder_emc_config.xml
```

For VMAX and VNX FC drivers, add the following entries:

```
volume_driver = cinder.volume.drivers.emc.emc_smis_fc.EMCSMISFCDriver
cinder_emc_config_file = /etc/cinder/cinder_emc_config.xml
```

Restart the `cinder-volume` service.

cinder_emc_config.xml configuration file

Create the `/etc/cinder/cinder_emc_config.xml` file. You do not need to restart the service for this change.

For VMAX, add the following lines to the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EMC>
  <StorageType>xxxx</StorageType>
  <MaskingView>xxxx</MaskingView>
  <EcomServerIp>x.x.x.x</EcomServerIp>
  <EcomServerPort>xxxx</EcomServerPort>
  <EcomUserName>xxxxxxxx</EcomUserName>
  <EcomPassword>xxxxxxxx</EcomPassword>
  <Timeout>xx</Timeout>
</EMC>
```

For VNX, add the following lines to the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EMC>
  <StorageType>xxxx</StorageType>
  <EcomServerIp>x.x.x.x</EcomServerIp>
  <EcomServerPort>xxxx</EcomServerPort>
  <EcomUserName>xxxxxxxx</EcomUserName>
  <EcomPassword>xxxxxxxx</EcomPassword>
  <Timeout>xx</Timeout>
</EMC>
```

Where:

- `StorageType` is the thin pool from which the user wants to create the volume. Thin pools can be created using Unisphere for VMAX and VNX. If the `StorageType` tag is not defined, you have to define volume types and set the pool name in extra specs.
- `EcomServerIp` and `EcomServerPort` are the IP address and port number of the ECOM server which is packaged with SMI-S.
- `EcomUserName` and `EcomPassword` are credentials for the ECOM server.
- `Timeout` specifies the maximum number of seconds you want to wait for an operation to finish.



Note

To attach VMAX volumes to an OpenStack VM, you must create a Masking View by using Unisphere for VMAX. The Masking View must have an Initiator Group that contains the initiator of the OpenStack compute node that hosts the VM.

Volume type support

Volume type support enables a single instance of `cinder-volume` to support multiple pools and thick/thin provisioning.

When the `StorageType` tag in `cinder_emc_config.xml` is used, the pool name is specified in the tag. Only thin provisioning is supported in this case.

When the `StorageType` tag is not used in `cinder_emc_config.xml`, the volume type needs to be used to define a pool name and a provisioning type. The pool name is the name of a pre-created pool. The provisioning type could be either `thin` or `thick`.

Here is an example of how to set up volume type. First create volume types. Then define extra specs for each volume type.

Procedure 1.5. Setup volume types

1. Create the volume types:

```
$ cinder type-create "High Performance"
$ cinder type-create "Standard Performance"
```

2. Setup the volume type extra specs:

```
$ cinder type-key "High Performance" set storagetype:pool=smi_pool
$ cinder type-key "High Performance" set storagetype:provisioning=thick
$ cinder type-key "Standard Performance" set storagetype:pool=smi_pool2
$ cinder type-key "Standard Performance" set storagetype:provisioning=thin
```

In the above example, two volume types are created. They are High Performance and Standard Performance. For High Performance, `storagetype:pool` is set to `smi_pool` and `storagetype:provisioning` is set to `thick`. Similarly for Standard Performance, `storagetype:pool` is set to `smi_pool2` and `storagetype:provisioning` is set to `thin`. If `storagetype:provisioning` is not specified, it will default to `thin`.

**Note**

Volume type names High Performance and Standard Performance are user-defined and can be any names. Extra spec keys `storagetype:pool` and `storagetype:provisioning` have to be the exact names listed here. Extra spec value `smi_pool` is your pool name. The extra spec value for `storagetype:provisioning` has to be either `thick` or `thin`. The driver will look for a volume type first. If the volume type is specified when creating a volume, the driver will look for the volume type definition and find the matching pool and provisioning type. If the volume type is not specified, it will fall back to use the `StorageType` tag in `cinder_emc_config.xml`.

GlusterFS driver

GlusterFS is an open-source scalable distributed file system that is able to grow to petabytes and beyond in size. More information can be found on [Gluster's homepage](#).

This driver enables use of GlusterFS in a similar fashion as the NFS driver. It supports basic volume operations, and like NFS, does not support snapshot/clone.

**Note**

You must use a Linux kernel of version 3.4 or greater (or version 2.6.32 or greater in Red Hat Enterprise Linux/CentOS 6.3+) when working with Gluster-based volumes. See [Bug 1177103](#) for more information.

To use Block Storage with GlusterFS, first set the `volume_driver` in `cinder.conf`:

```
volume_driver=cinder.volume.drivers.glusterfs.GlusterfsDriver
```

The following table contains the configuration options supported by the GlusterFS driver.

Table 1.4. Description of configuration options for storage_glusterfs

Configuration option = Default value	Description
[DEFAULT]	
<code>glusterfs_mount_point_base = \$state_path/mnt</code>	(StrOpt) Base dir containing mount points for gluster shares.

Configuration option = Default value	Description
glusterfs_qcow2_volumes = False	(BoolOpt) Create volumes as QCOW2 files rather than raw files.
glusterfs_shares_config = /etc/cinder/glusterfs_shares	(StrOpt) File with the list of available gluster shares
glusterfs_sparsed_volumes = True	(BoolOpt) Create volumes as sparsed files which take no space. If set to False volume is created as regular file. In such case volume creation takes a lot of time.

HDS HUS iSCSI driver

This Block Storage volume driver provides iSCSI support for [HUS \(Hitachi Unified Storage\)](#) arrays such as, HUS-110, HUS-130, and HUS-150.

System requirements

Use the HDS **hus-cmd** command to communicate with an HUS array. You can download this utility package from the HDS support site (<https://HDSSupport.hds.com>).

Platform: Ubuntu 12.04LTS or newer.

Supported operations

These operations are supported:

- Create volume
- Delete volume
- Attach volume
- Detach volume
- Clone volume
- Extend volume
- Create snapshot
- Delete snapshot
- Copy image to volume
- Copy volume to image
- Create volume from snapshot
- Get statistics

Configuration

The HDS driver supports the concept of differentiated services, where a volume type can be associated with the fine-tuned performance characteristics of an HDP— the dynamic pool

where volumes are created¹. For instance, an HDP can consist of fast SSDs to provide speed. HDP can provide a certain reliability based on things like its RAID level characteristics. HDS driver maps volume type to the `volume_type` option in its configuration file.

Configuration is read from an XML-format file. Examples are shown for single and multi back-end cases.



Note

- Configuration is read from an XML file. This example shows the configuration for single back-end and for multi-back-end cases.
- It is not recommended to manage an HUS array simultaneously from multiple OpenStack Block Storage instances or servers.²

Table 1.5. Description of configuration options for hds-hus

Configuration option = Default value	Description
[DEFAULT]	
<code>hds_cinder_config_file = /opt/hds/hus/cinder_hus_conf.xml</code>	(StrOpt) configuration file for HDS cinder plugin for HUS

HUS setup

Before using iSCSI services, use the HUS UI to create an iSCSI domain for each EVS providing iSCSI services.

Single back-end

In a single back-end deployment, only one OpenStack Block Storage instance runs on the OpenStack Block Storage server and controls one HUS array: this deployment requires these configuration files:

1. Set the `hds_cinder_config_file` option in the `/etc/cinder/cinder.conf` file to use the HDS volume driver. This option points to a configuration file.³

```
volume_driver = cinder.volume.drivers.hds.hds.HUSDriver
hds_cinder_config_file = /opt/hds/hus/cinder_hds_conf.xml
```

2. Configure `hds_cinder_config_file` at the location specified previously. For example, `/opt/hds/hus/cinder_hds_conf.xml`:

```
<?xml version="1.0" encoding="UTF-8" ?>
<config>
  <mgmt_ip0>172.17.44.16</mgmt_ip0>
  <mgmt_ip1>172.17.44.17</mgmt_ip1>
  <hus_cmd>hus-cmd</hus_cmd>
  <username>system</username>
  <password>manager</password>
  <svc_0>
    <volume_type>default</volume_type>
```

¹Do not confuse differentiated services with the OpenStack Block Storage volume services.

²It is okay to manage multiple HUS arrays by using multiple OpenStack Block Storage instances (or servers).

³The configuration file location may differ.

exactly match. If no configured `volume_types` match the incoming requested type, an error occurs in volume creation.

2. HDP, the pool ID associated with the service.
3. An iSCSI port dedicated to the service.

Typically a OpenStack Block Storage volume instance has only one such service label. For example, any `svc_0`, `svc_1`, `svc_2`, or `svc_3` can be associated with it. But any mix of these service labels can be used in the same instance⁵.

Table 1.6. Configuration options

Option	Type	Default	Description
<code>mgmt_ip0</code>	Required		Management Port 0 IP address
<code>mgmt_ip1</code>	Required		Management Port 1 IP address
<code>hus_cmd</code>	Optional		<code>hus_cmd</code> is the command used to communicate with the HUS array. If it is not set, the default value is <code>hus-cmd</code> .
<code>username</code>	Optional		Username is required only if secure mode is used
<code>password</code>	Optional		Password is required only if secure mode is used
<code>svc_0</code> , <code>svc_1</code> , <code>svc_2</code> , <code>svc_3</code>	Optional	(at least one label has to be defined)	Service labels: these four predefined names help four different sets of configuration options – each can specify iSCSI port address, HDP and a unique volume type.
<code>snapshot</code>	Required		A service label which helps specify configuration for snapshots, such as, HDP.
<code>volume_type</code>	Required		<code>volume_type</code> tag is used to match volume type. Default meets any type of <code>volume_type</code> , or if it is not specified. Any other <code>volume_type</code> is selected if exactly matched during <code>create_volume</code> .
<code>iscsi_ip</code>	Required		iSCSI port IP address where volume attaches for this volume type.
<code>hdp</code>	Required		HDP, the pool number where volume, or snapshot should be created.
<code>lun_start</code>	Optional	0	LUN allocation starts at this number.
<code>lun_end</code>	Optional	4096	LUN allocation is up to, but not including, this number.

HP 3PAR Fibre Channel and iSCSI drivers

The `HP3PARFCDriver` and `HP3PARISCSIDriver` drivers, which are based on the Block Storage service (Cinder) plug-in architecture, run volume operations by communicating with the HP 3PAR storage system over HTTP, HTTPS, and SSH connections. The HTTP and HTTPS communications use `hp3parclient`, which is part of the Python standard library.

For information about how to manage HP 3PAR storage systems, see the HP 3PAR user documentation.

System requirements

To use the HP 3PAR drivers, install the following software and components on the HP 3PAR storage system:

⁵The `get_volume_stats()` always provides the available capacity based on the combined sum of all the HDPs that are used in these services labels.

- HP 3PAR Operating System software version 3.1.3 or higher
- HP 3PAR Web Services API Server must be enabled and running
- One Common Provisioning Group (CPG)
- Additionally, you must install the `hp3parclient` version 3.0 or newer from the Python standard library on the system with the enabled Block Storage service volume drivers.

Supported operations

- Create volumes.
- Delete volumes.
- Extend volumes.
- Attach volumes.
- Detach volumes.
- Create snapshots.
- Delete snapshots.
- Create volumes from snapshots.
- Create cloned volumes.
- Copy images to volumes.
- Copy volumes to images.
- Volume migration(backend assisted).

Volume type support for both HP 3PAR drivers includes the ability to set the following capabilities in the OpenStack Block Storage API `cinder.api.contrib.types_extra_specs` volume type extra specs extension module:

- `hp3par:cpg`
- `hp3par:snap_cpg`
- `hp3par:provisioning`
- `hp3par:persona`
- `hp3par:vvs`

To work with the default filter scheduler, the key values are case sensitive and scoped with `hp3par:.` For information about how to set the key-value pairs and associate them with a volume type, run the following command:

```
$ cinder help type-key
```



Note

Volumes that are cloned only support extra specs keys `cpg`, `snap_cpg`, `provisioning` and `vvs`. The others are ignored. In addition the comments section of the cloned volume in the HP 3PAR StoreServ storage array is not populated.

If volume types are not used or a particular key is not set for a volume type, the following defaults are used:

- `hp3par:cpg` - Defaults to the `hp3par_cpg` setting in the `cinder.conf` file.
- `hp3par:snap_cpg` - Defaults to the `hp3par_snap` setting in the `cinder.conf` file. If `hp3par_snap` is not set, it defaults to the `hp3par_cpg` setting.
- `hp3par:provisioning` - Defaults to thin provisioning, the valid values are `thin` and `full`.
- `hp3par:persona` - Defaults to the `1 - Generic persona`. The valid values are, `1 - Generic`, `2 - Generic-ALUA`, `6 - Generic-legacy`, `7 - HPUX-legacy`, `8 - AIX-legacy`, `9 - EGENERA`, `10 - ONTAP-legacy`, `11 - VMware`, `12 - OpenVMS`, `13 - HPUX`, and `15 - WindowsServer`.

QoS support for both HP 3PAR drivers includes the ability to set the following capabilities in the OpenStack Block Storage API `cinder.api.contrib.qos_specs_manage qos specs` extension module:

- `minBWS`
- `maxBWS`
- `minIOPS`
- `maxIOPS`
- `latency`
- `priority`

The `qos` keys above no longer require to be scoped but must be created and associated to a volume type. For information about how to set the key-value pairs and associate them with a volume type, run the following commands:

```
$ cinder help qos-create
```

```
$ cinder help qos-key
```

```
$ cinder help qos-associate
```

The following keys require that the HP 3PAR StoreServ storage array has a Priority Optimization license installed.

- `hp3par:vvs` - The virtual volume set name that has been predefined by the Administrator with Quality of Service (QoS) rules associated to it. If you specify extra_specs `hp3par:vvs`, the `qos_specs minIOPS`, `maxIOPS`, `minBWS`, and `maxBWS` settings are ignored.

- `minBWS` - The QoS I/O issue bandwidth minimum goal in MBs. If not set, the I/O issue bandwidth rate has no minimum goal.
- `maxBWS` - The QoS I/O issue bandwidth rate limit in MBs. If not set, the I/O issue bandwidth rate has no limit.
- `minIOPS` - The QoS I/O issue count minimum goal. If not set, the I/O issue count has no minimum goal.
- `maxIOPS` - The QoS I/O issue count rate limit. If not set, the I/O issue count rate has no limit.
- `latency` - The latency goal in milliseconds.
- `priority` - The priority of the QoS rule over other rules. If not set, the priority is normal, valid values are low, normal and high.



Note

Since the Icehouse release, `minIOPS` and `maxIOPS` must be used together to set I/O limits. Similarly, `minBWS` and `maxBWS` must be used together. If only one is set the other will be set to the same value.

Enable the HP 3PAR Fibre Channel and iSCSI drivers

The `HP3PARFCDriver` and `HP3PARISCSIDriver` are installed with the OpenStack software.

1. Install the `hp3parclient` Python package on the OpenStack Block Storage system.

```
# pip install 'hp3parclient>=3.0,<4.0'
```

2. Verify that the HP 3PAR Web Services API server is enabled and running on the HP 3PAR storage system.

- a. Log onto the HP 3PAR storage system with administrator access.

```
$ ssh 3paradm@<HP 3PAR IP Address>
```

- b. View the current state of the Web Services API Server.

```
# showsapi
-Service- -State- -HTTP_State- HTTP_Port -HTTPS_State- HTTPS_Port -
Version-
Enabled   Active Enabled           8008           Enabled       8080
1.1
```

- c. If the Web Services API Server is disabled, start it.

```
# startwsapi
```

3. If the HTTP or HTTPS state is disabled, enable one of them.

```
# setwsapi -http enable
```

or

```
# setwsapi -https enable
```

**Note**

To stop the Web Services API Server, use the `stopwsapi` command. For other options run the `setwsapi -h` command.

4. If you are not using an existing CPG, create a CPG on the HP 3PAR storage system to be used as the default location for creating volumes.
5. Make the following changes in the `/etc/cinder/cinder.conf` file.

```
## REQUIRED SETTINGS
# 3PAR WS API Server URL
hp3par_api_url=https://10.10.0.141:8080/api/v1

# 3PAR Super user username
hp3par_username=3paradm

# 3PAR Super user password
hp3par_password=3parpass

# 3PAR CPG to use for volume creation
hp3par_cpg=OpenStackCPG_RAID5_NL

# IP address of SAN controller for SSH access to the array
san_ip=10.10.22.241

# Username for SAN controller for SSH access to the array
san_login=3paradm

# Password for SAN controller for SSH access to the array
san_password=3parpass

# FIBRE CHANNEL(uncomment the next line to enable the FC driver)
# volume_driver=cinder.volume.drivers.san.hp.hp_3par_fc.HP3PARFCDriver

# iSCSI (uncomment the next line to enable the iSCSI driver and
# hp3par_iscsi_ips or iscsi_ip_address)
#volume_driver=cinder.volume.drivers.san.hp.hp_3par_iscsi.
HP3PARISCSIDriver

# iSCSI multiple port configuration
# hp3par_iscsi_ips=10.10.220.253:3261,10.10.222.234

# Still available for single port iSCSI configuration
#iscsi_ip_address=10.10.220.253

## OPTIONAL SETTINGS
# Enable HTTP debugging to 3PAR
hp3par_debug=False

# The CPG to use for Snapshots for volumes. If empty hp3par_cpg will be
used.
hp3par_snap_cpg=OpenStackSNAP_CPG

# Time in hours to retain a snapshot. You can't delete it before this
expires.
hp3par_snapshot_retention=48
```

```
# Time in hours when a snapshot expires and is deleted. This must be  
larger than retention.  
hp3par_snapshot_expiration=72
```



Note

You can enable only one driver on each cinder instance unless you enable multiple back-end support. See the Cinder multiple back-end support instructions to enable this feature.



Note

You can configure one or more iSCSI addresses by using the `hp3par_iscsi_ips` option. When you configure multiple addresses, the driver selects the iSCSI port with the fewest active volumes at attach time. The IP address might include an IP port by using a colon (:) to separate the address from port. If you do not define an IP port, the default port 3260 is used. Separate IP addresses with a comma (,). The `iscsi_ip_address/iscsi_port` options might be used as an alternative to `hp3par_iscsi_ips` for single port iSCSI configuration.

6. Save the changes to the `cinder.conf` file and restart the `cinder-volume` service.

The HP 3PAR Fibre Channel and iSCSI drivers are now enabled on your OpenStack system. If you experience problems, review the Block Storage service log files for errors.

HP LeftHand/StoreVirtual driver

The `HPLeftHandISCSIDriver` is based on the Block Storage service (Cinder) plugin architecture. Volume operations are run by communicating with the HP LeftHand/StoreVirtual system over HTTPS, or SSH connections. HTTPS communications use the `hplefthandclient`, which is part of the Python standard library.

The `HPLeftHandISCSIDriver` can be configured to run in one of two possible modes, legacy mode which uses SSH/CLIQ to communicate with the HP LeftHand/StoreVirtual array, or standard mode which uses a new REST client to communicate with the array. No new functionality has been, or will be, supported in legacy mode. For performance improvements and new functionality, the driver must be configured for standard mode, the `hplefthandclient` must be downloaded, and HP LeftHand/StoreVirtual Operating System software version 11.5 or higher is required on the array. To configure the driver in standard mode, see [the section called "HP LeftHand/StoreVirtual REST driver standard mode" \[29\]](#). To configure the driver in legacy mode, see [the section called "HP LeftHand/StoreVirtual CLIQ driver legacy mode" \[32\]](#).

For information about how to manage HP LeftHand/StoreVirtual storage systems, see the HP LeftHand/StoreVirtual user documentation.

HP LeftHand/StoreVirtual REST driver standard mode

This section describes how to configure the HP LeftHand/StoreVirtual Cinder driver in standard mode.

System requirements

To use the HP LeftHand/StoreVirtual driver in standard mode, do the following:

- Install LeftHand/StoreVirtual Operating System software version 11.5 or higher on the HP LeftHand/StoreVirtual storage system.
- Create a cluster group.
- Install the `hplefthandclient` version 1.0.2 from the Python Package Index on the system with the enabled Block Storage service volume drivers.

Supported operations

- Create volumes
- Delete volumes
- Extend volumes
- Attach volumes
- Detach volumes
- Create snapshots
- Delete snapshots
- Create volumes from snapshots
- Create cloned volumes
- Copy images to volumes
- Copy volumes to images
- Backend assisted volume migration
- Volume retype

When a volume is migrated, using Backend assisted volume migration, both source and destination clusters must be in the same HP LeftHand/StoreVirtual management group. The HP LeftHand/StoreVirtual array will use native LeftHand APIs to migrate the volume. The volume cannot be attached or have snapshots to migrate.

Volume type support for the driver includes the ability to set the following capabilities in the OpenStack Cinder API `cinder.api.contrib.types_extra_specs` volume type extra specs extension module.

- `hplh:provisioning`
- `hplh:ao`
- `hplh:data_pl`

To work with the default filter scheduler, the key values are case sensitive and scoped with 'hplh: '. For information about how to set the key-value pairs and associate them with a volume type, run the following command:

```
$ cinder help type-key
```

- The following keys require that the HP LeftHand/StoreVirtual storage array be configured for.

hplh:ao The HP LeftHand/StoreVirtual storage array must be configured for Adaptive Optimization.

hplh:data_pl The HP LeftHand/StoreVirtual storage array must be able to support the Data Protection level specified by the extra spec.

- If volume types are not used or a particular key is not set for a volume type, the following defaults are used:

hplh:provisioning Defaults to *thin* provisioning, the valid values are, *thin* and *full*

hplh:ao Defaults to *true*, the valid values are, *true* and *false*.

hplh:data_pl Defaults to *r-0*, Network RAID-0 (None), the valid values are,
r-0, Network RAID-0 (None)
r-5, Network RAID-5 (Single Parity)
r-10-2, Network RAID-10 (2-Way Mirror)
r-10-3, Network RAID-10 (3-Way Mirror)
r-10-4, Network RAID-10 (4-Way Mirror)
r-6, Network RAID-6 (Dual Parity),

Enable the HP LeftHand/StoreVirtual iSCSI driver in standard mode

The `HPLeftHandISCSIDriver` is installed with the OpenStack software.

1. Install the `hplefthandclient` Python package on the OpenStack Block Storage system.

```
# pip install 'hplefthandclient>=1.0.2,<2.0'
```

2. If you are not using an existing cluster, create a cluster on the HP LeftHand storage system to be used as the cluster for creating volumes.
3. Make the following changes in the `/etc/cinder/cinder.conf` file:

```
## REQUIRED SETTINGS

# LeftHand WS API Server URL
hplefthand_api_url=https://10.10.0.141:8081/lhos
```

```
# LeftHand Super user username
hplefthand_username=lhuser

# LeftHand Super user password
hplefthand_password=lhpass

# LeftHand cluster to use for volume creation
hplefthand_clustername=ClusterLefthand

# LeftHand iSCSI driver
volume_driver=cinder.volume.drivers.san.hp.hp_lefthand_iscsi.
HPLefthandISCSIDriver

## OPTIONAL SETTINGS

# Should CHAPS authentication be used (default=false)
hplefthand_iscsi_chap_enabled=false

# Enable HTTP debugging to LeftHand (default=false)
hplefthand_debug=false
```

You can enable only one driver on each cinder instance unless you enable multiple back-end support. See the Cinder multiple back-end support instructions to enable this feature.

If the `hplefthand_iscsi_chap_enabled` is set to `true`, the driver will associate randomly-generated CHAP secrets with all hosts on the HP LeftHand/StoreVirtual system. OpenStack Compute nodes use these secrets when creating iSCSI connections.



Important

CHAP secrets are passed from OpenStack Block Storage to Compute in clear text. This communication should be secured to ensure that CHAP secrets are not discovered.



Note

CHAP secrets are added to existing hosts as well as newly-created ones. If the CHAP option is enabled, hosts will not be able to access the storage without the generated secrets.

4. Save the changes to the `cinder.conf` file and restart the `cinder-volume` service.

The HP LeftHand/StoreVirtual driver is now enabled in standard mode on your OpenStack system. If you experience problems, review the Block Storage service log files for errors.

HP LeftHand/StoreVirtual CLIQ driver legacy mode

This section describes how to configure the HP LeftHand/StoreVirtual Cinder driver in legacy mode.

The `HPLefthandISCSIDriver` allows you to use a HP Lefthand/StoreVirtual SAN that supports the CLIQ interface. Every supported volume operation translates into a CLIQ call in the back-end.

Supported operations

- Create volumes
- Delete volumes
- Extend volumes
- Attach volumes
- Detach volumes
- Create snapshots.
- Delete snapshots
- Create volumes from snapshots
- Copy images to volumes
- Copy volumes to images

Enable the HP LeftHand/StoreVirtual iSCSI driver in legacy mode

The `HPLeftHandISCSIDriver` is installed with the OpenStack software.

1. If you are not using an existing cluster, create a cluster on the HP Lefthand storage system to be used as the cluster for creating volumes.
2. Make the following changes in the `/etc/cinder/cinder.conf` file.

```
## REQUIRED SETTINGS

# VIP of your Virtual Storage Appliance (VSA).
san_ip=10.10.0.141

# LeftHand Super user username
san_login=lhuser

# LeftHand Super user password
san_password=lhpass

# LeftHand ssh port, the default for the VSA is usually 16022.
san_ssh_port=16022

# LeftHand cluster to use for volume creation
san_clustername=ClusterLefthand

# LeftHand iSCSI driver
volume_driver=cinder.volume.drivers.san.hp.hp_lefthand_iscsi.
HPLeftHandISCSIDriver

## OPTIONAL SETTINGS

# LeftHand provisioning, to disable thin provisioning, set to
# set to False.
```

```
san_thin_provision=True

# Typically, this parameter is set to False, for this driver.
# To configure the CLIQ commands to run locally instead of over ssh,
# set this parameter to True
san_is_local=False
```

3. Save the changes to the `cinder.conf` file and restart the `cinder-volume` service.

The HP LeftHand/StoreVirtual driver is now enabled in legacy mode on your OpenStack system. If you experience problems, review the Block Storage service log files for errors.

To configure the VSA

1. Configure CHAP on each of the `nova-compute` nodes.
2. Add server associations on the VSA with the associated CHAPS and initiator information. The name should correspond to the `hostname` of the `nova-compute` node. For Xen, this is the hypervisor host name. To do this, use either CLIQ or the Centralized Management Console.

HP MSA Fibre Channel driver

The HP MSA fiber channel driver runs volume operations on the storage array over HTTP.

A VDisk must be created on the HP MSA array first. This can be done using the web interface or the command-line interface of the array.

The following options must be defined in the `cinder-volume` configuration file (`/etc/cinder/cinder.conf`):

- Set the `volume_driver` option to `cinder.volume.drivers.san.hp.hp_msa_fc.HPMSAFCDriver`
- Set the `san_ip` option to the hostname or IP address of your HP MSA array.
- Set the `san_login` option to the login of an existing user of the HP MSA array.
- Set the `san_password` option to the password for this user.

Huawei storage driver

The Huawei driver supports the iSCSI and Fibre Channel connections and enables OceanStor T series unified storage and OceanStor 18000 high-end storage to provide block storage services for OpenStack.

Supported operations

OceanStor T series unified storage supports these operations:

- Create, delete, attach, and detach volumes.

- Create, list, and delete volume snapshots.
- Create a volume from a snapshot.
- Copy an image to a volume.
- Copy a volume to an image.
- Clone a volume.

OceanStor 18000 supports these operations:

- Create, delete, attach, and detach volumes.
- Create, list, and delete volume snapshots.
- Copy an image to a volume.
- Copy a volume to an image.
- Create a volume from a snapshot.
- Clone a volume.

Configure Block Storage nodes

In `/etc/cinder`, create the driver configuration file named `cinder_huawei_conf.xml`.

You must configure `Product` and `Protocol` to specify a storage system and link type. The following uses the iSCSI driver as an example. The driver configuration file of OceanStor T series unified storage is shown as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    <Product>T</Product>
    <Protocol>iSCSI</Protocol>
    <ControllerIP0>x.x.x.x</ControllerIP0>
    <ControllerIP1>x.x.x.x</ControllerIP1>
    <UserName>xxxxxxx</UserName>
    <UserPassword>xxxxxxx</UserPassword>
  </Storage>
  <LUN>
    <LUNType>Thick</LUNType>
    <StripUnitSize>64</StripUnitSize>
    <WriteType>1</WriteType>
    <MirrorSwitch>1</MirrorSwitch>
    <Prefetch Type="3" value="0" />
    <StoragePool Name="xxxxxxx" />
    <StoragePool Name="xxxxxxx" />
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
    <Initiator Name="xxxxxxx" TargetIP="x.x.x.x" />
  </iSCSI>
</config>
```

```
<Initiator Name="xxxxxxx" TargetIP="x.x.x.x"/>
</iSCSI>
<Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x"/>
</config>
```

The driver configuration file of OceanStor 18000 is shown as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    <Product>18000</Product>
    <Protocol>iSCSI</Protocol>
    <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
    <UserName>xxxxxxx</UserName>
    <UserPassword>xxxxxxx</UserPassword>
  </Storage>
  <LUN>
    <LUNType>Thick</LUNType>
    <WriteType>1</WriteType>
    <MirrorSwitch>1</MirrorSwitch>
    <StoragePool>xxxxxxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
    <Initiator Name="xxxxxxx" TargetIP="x.x.x.x"/>
    <Initiator Name="xxxxxxx" TargetIP="x.x.x.x"/>
  </iSCSI>
  <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x"/>
</config>
```



Note for Fibre Channel driver configuration

You do not need to configure the iSCSI target IP address for the Fibre Channel driver. In the prior example, delete the iSCSI configuration:

```
<iSCSI>
  <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  <Initiator Name="xxxxxxx" TargetIP="x.x.x.x"/>
  <Initiator Name="xxxxxxx" TargetIP="x.x.x.x"/>
</iSCSI>
```

To add `volume_driver` and `cinder_huawei_conf_file` items, you can modify the `cinder.conf` configuration file as follows:

```
volume_driver = cinder.volume.drivers.huawei.HuaweiVolumeDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
```

You can configure multiple Huawei back-end storages as follows:

```
enabled_backends = t_iscsi, 18000_iscsi
[t_iscsi]
volume_driver = cinder.volume.drivers.huawei.HuaweiVolumeDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_t_iscsi.xml
volume_backend_name = HuaweiTISCSIDriver
[18000_iscsi]
volume_driver = cinder.volume.drivers.huawei.HuaweiVolumeDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_18000_iscsi.xml
volume_backend_name = Huawei18000ISCSIDriver
```

Configuration file details

This table describes the Huawei storage driver configuration options:

Table 1.7. Huawei storage driver configuration options

Flag name	Type	Default	Description
Product	Required		Type of a storage product. Valid values are Tor 18000.
Protocol	Required		Type of a protocol. Valid values are iSCSI or FC.
ControllerIP0	Required		IP address of the primary controller (not required for the 18000)
ControllerIP1	Required		IP address of the secondary controller (not required for the 18000)
RestURL	Required		Access address of the Rest port (required only for the 18000)
UserName	Required		User name of an administrator
UserPassword	Required		Password of an administrator
LUNType	Optional	Thin	Type of a created LUN. Valid values are Thick or Thin.
StripUnitSize	Optional	64	Stripe depth of a created LUN. The value is expressed in KB. This flag is not valid for a thin LUN.
WriteType	Optional	1	Cache write method. The method can be write back, write through, or Required write back. The default value is 1, indicating write back.
MirrorSwitch	Optional	1	Cache mirroring policy. The default value is 1, indicating that a mirroring policy is used.
Prefetch Type	Optional	3	Cache prefetch strategy. The strategy can be constant prefetch, variable prefetch, or intelligent prefetch. Default value is 3, which indicates intelligent prefetch and is not required for the 18000.
Prefetch Value	Optional	0	Cache prefetch value.
StoragePool	Required		Name of a storage pool that you want to use.
DefaultTargetIP	Optional		Default IP address of the iSCSI port provided for compute nodes.
Initiator Name	Optional		Name of a compute node initiator.
Initiator TargetIP	Optional		IP address of the iSCSI port provided for compute nodes.
OSType	Optional	Linux	The OS type for a compute node.
HostIP	Optional		The IPs for compute nodes.



Note for the configuration

1. You can configure one iSCSI target port for each or all compute nodes. The driver checks whether a target port IP address is configured for the current compute node. If not, select `DefaultTargetIP`.

2. You can configure multiple storage pools in one configuration file, which supports the use of multiple storage pools in a storage system. (The 18000 driver allows configuration of only one storage pool.)
3. For details about LUN configuration information, see the **createlun** command in the command-line interface (CLI) documentation or run the **help -c createlun** on the storage system CLI.
4. After the driver is loaded, the storage system obtains any modification of the driver configuration file in real time and you do not need to restart the `cinder-volume` service.
5. The driver does not support the iSCSI multipath scenarios.

IBM XIV/DS8K volume driver

There is a unified volume back-end for IBM XIV and DS8K storage. Set the following in your `cinder.conf`, and use the following options to configure it.

```
volume_driver=cinder.volume.drivers.xiv_ds8k.XIVDS8KDriver
```

Table 1.8. Description of configuration options for xiv

Configuration option = Default value	Description
[DEFAULT]	
<code>xiv_chap = disabled</code>	(StrOpt) CHAP authentication mode, effective only for iscsi (disabled enabled)
<code>xiv_ds8k_connection_type = iscsi</code>	(StrOpt) Connection type to the IBM Storage Array (fibre_channel iscsi)
<code>xiv_ds8k_proxy = xiv_ds8k_openstack.nova_proxy.XIVDS8KNovaProxy</code>	(StrOpt) Proxy driver that connects to the IBM Storage Array

IBM GPFS volume driver

IBM General Parallel File System (GPFS) is a cluster file system that provides concurrent access to file systems from multiple nodes. The storage provided by these nodes can be direct attached, network attached, SAN attached, or a combination of these methods. GPFS provides many features beyond common data access, including data replication, policy based storage management, and space efficient file snapshot and clone operations.

How the GPFS driver works

The GPFS driver enables the use of GPFS in a fashion similar to that of the NFS driver. With the GPFS driver, instances do not actually access a storage device at the block level. Instead, volume backing files are created in a GPFS file system and mapped to instances, which emulate a block device.



Note

GPFS software must be installed and running on nodes where Block Storage and Compute services run in the OpenStack environment. A GPFS file system

must also be created and mounted on these nodes before starting the `cinder-volume` service. The details of these GPFS specific steps are covered in *GPFS: Concepts, Planning, and Installation Guide* and *GPFS: Administration and Programming Reference*.

Optionally, the Image Service can be configured to store images on a GPFS file system. When a Block Storage volume is created from an image, if both image data and volume data reside in the same GPFS file system, the data from image file is moved efficiently to the volume file using copy-on-write optimization strategy.

Enable the GPFS driver

To use the Block Storage service with the GPFS driver, first set the `volume_driver` in `cinder.conf`:

```
volume_driver = cinder.volume.drivers.ibm.gpfs.GPFSDriver
```

The following table contains the configuration options supported by the GPFS driver.

Table 1.9. Description of configuration options for `storage_gpfs`

Configuration option = Default value	Description
[DEFAULT]	
<code>gpfs_images_dir = None</code>	(StrOpt) Specifies the path of the Image service repository in GPFS. Leave undefined if not storing images in GPFS.
<code>gpfs_images_share_mode = None</code>	(StrOpt) Specifies the type of image copy to be used. Set this when the Image service repository also uses GPFS so that image files can be transferred efficiently from the Image service to the Block Storage service. There are two valid values: "copy" specifies that a full copy of the image is made; "copy_on_write" specifies that copy-on-write optimization strategy is used and unmodified blocks of the image file are shared efficiently.
<code>gpfs_max_clone_depth = 0</code>	(IntOpt) Specifies an upper limit on the number of indirections required to reach a specific block due to snapshots or clones. A lengthy chain of copy-on-write snapshots or clones can have a negative impact on performance, but improves space utilization. 0 indicates unlimited clone depth.
<code>gpfs_mount_point_base = None</code>	(StrOpt) Specifies the path of the GPFS directory where Block Storage volume and snapshot files are stored.
<code>gpfs_sparse_volumes = True</code>	(BoolOpt) Specifies that volumes are created as sparse files which initially consume no space. If set to False, the volume is created as a fully allocated file, in which case, creation may take a significantly longer time.
<code>gpfs_storage_pool = None</code>	(StrOpt) Specifies the storage pool that volumes are assigned to. By default, the system storage pool is used.



Note

The `gpfs_images_share_mode` flag is only valid if the Image Service is configured to use GPFS with the `gpfs_images_dir` flag. When the value of this flag is `copy_on_write`, the paths specified by the `gpfs_mount_point_base` and `gpfs_images_dir` flags must both reside in the same GPFS file system and in the same GPFS file set.

Volume creation options

It is possible to specify additional volume configuration options on a per-volume basis by specifying volume metadata. The volume is created using the specified options. Changing the metadata after the volume is created has no effect. The following table lists the volume creation options supported by the GPFS volume driver.

Table 1.10. Volume Create Options for GPFS Volume Drive

Metadata Item Name	Description
<code>fstype</code>	Specifies whether to create a file system or a swap area on the new volume. If <code>fstype=swap</code> is specified, the <code>mkswap</code> command is used to create a swap area. Otherwise the <code>mkfs</code> command is passed the specified file system type, for example <code>ext3</code> , <code>ext4</code> or <code>ntfs</code> .
<code>fslabel</code>	Sets the file system label for the file system specified by <code>fstype</code> option. This value is only used if <code>fstype</code> is specified.
<code>data_pool_name</code>	Specifies the GPFS storage pool to which the volume is to be assigned. Note: The GPFS storage pool must already have been created.
<code>replicas</code>	Specifies how many copies of the volume file to create. Valid values are 1, 2, and, for GPFS V3.5.0.7 and later, 3. This value cannot be greater than the value of the <code>MaxDataReplicas</code> attribute of the file system.
<code>dio</code>	Enables or disables the Direct I/O caching policy for the volume file. Valid values are <code>yes</code> and <code>no</code> .
<code>write_affinity_depth</code>	Specifies the allocation policy to be used for the volume file. Note: This option only works if <code>allow-write-affinity</code> is set for the GPFS data pool.
<code>block_group_factor</code>	Specifies how many blocks are laid out sequentially in the volume file to behave as a single large block. Note: This option only works if <code>allow-write-affinity</code> is set for the GPFS data pool.
<code>write_affinity_failure_group</code>	Specifies the range of nodes (in GPFS shared nothing architecture) where replicas of blocks in the volume file are to be written. See <i>GPFS: Administration and Programming Reference</i> for more details on this option.

Example: Volume creation options

This example shows the creation of a 50GB volume with an `ext4` file system labeled `newfs` and direct IO enabled:

```
$ cinder create --metadata fstype=ext4 fslabel=newfs dio=yes --display-name volume_1 50
```

Operational notes for GPFS driver

Snapshots and clones

Volume snapshots are implemented using the GPFS file clone feature. Whenever a new snapshot is created, the snapshot file is efficiently created as a read-only clone parent of the volume, and the volume file uses copy-on-write optimization strategy to minimize data movement.

Similarly when a new volume is created from a snapshot or from an existing volume, the same approach is taken. The same approach is also used when a new volume is created from an Image Service image, if the source image is in raw format, and `gpfs_images_share_mode` is set to `copy_on_write`.

IBM Storwize family and SVC volume driver

The volume management driver for Storwize family and SAN Volume Controller (SVC) provides OpenStack Compute instances with access to IBM Storwize family or SVC storage systems.

Configure the Storwize family and SVC system

Network configuration

The Storwize family or SVC system must be configured for iSCSI, Fibre Channel, or both.

If using iSCSI, each Storwize family or SVC node should have at least one iSCSI IP address. The IBM Storwize/SVC driver uses an iSCSI IP address associated with the volume's preferred node (if available) to attach the volume to the instance, otherwise it uses the first available iSCSI IP address of the system. The driver obtains the iSCSI IP address directly from the storage system; you do not need to provide these iSCSI IP addresses directly to the driver.



Note

If using iSCSI, ensure that the compute nodes have iSCSI network access to the Storwize family or SVC system.



Note

OpenStack Nova's Grizzly version supports iSCSI multipath. Once this is configured on the Nova host (outside the scope of this documentation), multipath is enabled.

If using Fibre Channel (FC), each Storwize family or SVC node should have at least one WWPN port configured. If the `storwize_svc_multipath_enabled` flag is set to `True` in the Cinder configuration file, the driver uses all available WWPNs to attach the volume to the instance (details about the configuration flags appear in the [next section](#)). If the flag is not set, the driver uses the WWPN associated with the volume's preferred node (if available), otherwise it uses the first available WWPN of the system. The driver obtains the WWPNs directly from the storage system; you do not need to provide these WWPNs directly to the driver.



Note

If using FC, ensure that the compute nodes have FC connectivity to the Storwize family or SVC system.

iSCSI CHAP authentication

If using iSCSI for data access and the `storwize_svc_iscsi_chap_enabled` is set to `True`, the driver will associate randomly-generated CHAP secrets with all hosts on the

Storwize family system. OpenStack compute nodes use these secrets when creating iSCSI connections.

**Note**

CHAP secrets are added to existing hosts as well as newly-created ones. If the CHAP option is enabled, hosts will not be able to access the storage without the generated secrets.

**Note**

Not all OpenStack Compute drivers support CHAP authentication. Please check compatibility before using.

**Note**

CHAP secrets are passed from OpenStack Block Storage to Compute in clear text. This communication should be secured to ensure that CHAP secrets are not discovered.

Configure storage pools

Each instance of the IBM Storwize/SVC driver allocates all volumes in a single pool. The pool should be created in advance and be provided to the driver using the `storwize_svc_volpool_name` configuration flag. Details about the configuration flags and how to provide the flags to the driver appear in the [next section](#).

Configure user authentication for the driver

The driver requires access to the Storwize family or SVC system management interface. The driver communicates with the management using SSH. The driver should be provided with the Storwize family or SVC management IP using the `san_ip` flag, and the management port should be provided by the `san_ssh_port` flag. By default, the port value is configured to be port 22 (SSH).

**Note**

Make sure the compute node running the `cinder-volume` management driver has SSH network access to the storage system.

To allow the driver to communicate with the Storwize family or SVC system, you must provide the driver with a user on the storage system. The driver has two authentication methods: password-based authentication and SSH key pair authentication. The user should have an Administrator role. It is suggested to create a new user for the management driver. Please consult with your storage and security administrator regarding the preferred authentication method and how passwords or SSH keys should be stored in a secure manner.

**Note**

When creating a new user on the Storwize or SVC system, make sure the user belongs to the Administrator group or to another group that has an Administrator role.

If using password authentication, assign a password to the user on the Storwize or SVC system. The driver configuration flags for the user and password are `san_login` and `san_password`, respectively.

If you are using the SSH key pair authentication, create SSH private and public keys using the instructions below or by any other method. Associate the public key with the user by uploading the public key: select the "choose file" option in the Storwize family or SVC management GUI under "SSH public key". Alternatively, you may associate the SSH public key using the command line interface; details can be found in the Storwize and SVC documentation. The private key should be provided to the driver using the `san_private_key` configuration flag.

Create a SSH key pair with OpenSSH

You can create an SSH key pair using OpenSSH, by running:

```
$ ssh-keygen -t rsa
```

The command prompts for a file to save the key pair. For example, if you select 'key' as the filename, two files are created: `key` and `key.pub`. The `key` file holds the private SSH key and `key.pub` holds the public SSH key.

The command also prompts for a pass phrase, which should be empty.

The private key file should be provided to the driver using the `san_private_key` configuration flag. The public key should be uploaded to the Storwize family or SVC system using the storage management GUI or command line interface.



Note

Ensure that Cinder has read permissions on the private key file.

Configure the Storwize family and SVC driver

Enable the Storwize family and SVC driver

Set the volume driver to the Storwize family and SVC driver by setting the `volume_driver` option in `cinder.conf` as follows:

```
volume_driver = cinder.volume.drivers.ibm.storwize_svc.StorwizeSVCDriver
```

Storwize family and SVC driver options in `cinder.conf`

The following options specify default values for all volumes. Some can be over-ridden using volume types, which are described below.

Table 1.11. List of configuration flags for Storwize storage and SVC driver

Flag name	Type	Default	Description
<code>san_ip</code>	Required		Management IP or host name
<code>san_ssh_port</code>	Optional	22	Management port
<code>san_login</code>	Required		Management login username
<code>san_password</code>	Required ^a		Management login password
<code>san_private_key</code>	Required ^a		Management login SSH private key

Flag name	Type	Default	Description
storwize_svc_volpool_name	Required		Default pool name for volumes
storwize_svc_vol_rsize	Optional	2	Initial physical allocation (percentage) ^b
storwize_svc_vol_warning	Optional	0 (disabled)	Space allocation warning threshold (percentage) ^b
storwize_svc_vol_autoexpand	Optional	True	Enable or disable volume auto expand ^c
storwize_svc_vol_grainsize	Optional	256	Volume grain size ^b in KB
storwize_svc_vol_compression	Optional	False	Enable or disable Real-time Compression ^d
storwize_svc_vol_easytier	Optional	True	Enable or disable Easy Tier ^e
storwize_svc_vol_iogrp	Optional	0	The I/O group in which to allocate vdisks
storwize_svc_flashcopy_timeout	Optional	120	FlashCopy timeout threshold ^f (seconds)
storwize_svc_connection_protocol	Optional	iSCSI	Connection protocol to use (currently supports 'iSCSI' or 'FC')
storwize_svc_iscsi_chap_enabled	Optional	True	Configure CHAP authentication for iSCSI connections
storwize_svc_multipath_enabled	Optional	False	Enable multipath for FC connections ^g
storwize_svc_multihost_enabled	Optional	True	Enable mapping vdisks to multiple hosts ^h

^aThe authentication requires either a password (`san_password`) or SSH private key (`san_private_key`). One must be specified. If both are specified, the driver uses only the SSH private key.

^bThe driver creates thin-provisioned volumes by default. The `storwize_svc_vol_rsize` flag defines the initial physical allocation percentage for thin-provisioned volumes, or if set to `-1`, the driver creates full allocated volumes. More details about the available options are available in the Storwize family and SVC documentation.

^cDefines whether thin-provisioned volumes can be auto expanded by the storage system, a value of `True` means that auto expansion is enabled, a value of `False` disables auto expansion. Details about this option can be found in the `-autoexpand` flag of the Storwize family and SVC command line interface `mkvdisk` command.

^dDefines whether Real-time Compression is used for the volumes created with OpenStack. Details on Real-time Compression can be found in the Storwize family and SVC documentation. The Storwize or SVC system must have compression enabled for this feature to work.

^eDefines whether Easy Tier is used for the volumes created with OpenStack. Details on EasyTier can be found in the Storwize family and SVC documentation. The Storwize or SVC system must have Easy Tier enabled for this feature to work.

^fThe driver wait timeout threshold when creating an OpenStack snapshot. This is actually the maximum amount of time that the driver waits for the Storwize family or SVC system to prepare a new FlashCopy mapping. The driver accepts a maximum wait time of 600 seconds (10 minutes).

^gMultipath for iSCSI connections requires no storage-side configuration and is enabled if the compute host has multipath configured.

^hThis option allows the driver to map a vdisk to more than one host at a time. This scenario occurs during migration of a virtual machine with an attached volume; the volume is simultaneously mapped to both the source and destination compute hosts. If your deployment does not require attaching vdisks to multiple hosts, setting this flag to `False` will provide added safety.

Table 1.12. Description of configuration options for storwize

Configuration option = Default value	Description
[DEFAULT]	
<code>storwize_svc_connection_protocol = iSCSI</code>	(StrOpt) Connection protocol (iSCSI/FC)
<code>storwize_svc_flashcopy_timeout = 120</code>	(IntOpt) Maximum number of seconds to wait for FlashCopy to be prepared. Maximum value is 600 seconds (10 minutes)
<code>storwize_svc_iscsi_chap_enabled = True</code>	(BoolOpt) Configure CHAP authentication for iSCSI connections (Default: Enabled)
<code>storwize_svc_multihostmap_enabled = True</code>	(BoolOpt) Allows vdisk to multi host mapping
<code>storwize_svc_multipath_enabled = False</code>	(BoolOpt) Connect with multipath (FC only; iSCSI multipath is controlled by Nova)
<code>storwize_svc_vol_autoexpand = True</code>	(BoolOpt) Storage system autoexpand parameter for volumes (True/False)
<code>storwize_svc_vol_compression = False</code>	(BoolOpt) Storage system compression option for volumes

Configuration option = Default value	Description
<code>storwize_svc_vol_easytier = True</code>	(BoolOpt) Enable Easy Tier for volumes
<code>storwize_svc_vol_grainsize = 256</code>	(IntOpt) Storage system grain size parameter for volumes (32/64/128/256)
<code>storwize_svc_vol_iogrp = 0</code>	(IntOpt) The I/O group in which to allocate volumes
<code>storwize_svc_vol_rsize = 2</code>	(IntOpt) Storage system space-efficiency parameter for volumes (percentage)
<code>storwize_svc_vol_warning = 0</code>	(IntOpt) Storage system threshold for volume capacity warnings (percentage)
<code>storwize_svc_volpool_name = volpool</code>	(StrOpt) Storage system storage pool for volumes

Placement with volume types

The IBM Storwize/SVC driver exposes capabilities that can be added to the `extra_specs` of volume types, and used by the filter scheduler to determine placement of new volumes. Make sure to prefix these keys with `capabilities:` to indicate that the scheduler should use them. The following `extra_specs` are supported:

- `capabilities:volume_back-end_name` - Specify a specific back-end where the volume should be created. The back-end name is a concatenation of the name of the IBM Storwize/SVC storage system as shown in `lssystem`, an underscore, and the name of the pool (mdisk group). For example:

```
capabilities:volume_back-end_name=myV7000_openstackpool
```

- `capabilities:compression_support` - Specify a back-end according to compression support. A value of `True` should be used to request a back-end that supports compression, and a value of `False` will request a back-end that does not support compression. If you do not have constraints on compression support, do not set this key. Note that specifying `True` does not enable compression; it only requests that the volume be placed on a back-end that supports compression. Example syntax:

```
capabilities:compression_support='<is> True'
```

- `capabilities:easytier_support` - Similar semantics as the `compression_support` key, but for specifying according to support of the Easy Tier feature. Example syntax:

```
capabilities:easytier_support='<is> True'
```

- `capabilities:storage_protocol` - Specifies the connection protocol used to attach volumes of this type to instances. Legal values are `iSCSI` and `FC`. This `extra_specs` value is used for both placement and setting the protocol used for this volume. In the example syntax, note `<in>` is used as opposed to `<is>` used in the previous examples.

```
capabilities:storage_protocol='<in> FC'
```

Configure per-volume creation options

Volume types can also be used to pass options to the IBM Storwize/SVC driver, which override the default values set in the configuration file. Contrary to the previous examples where the "capabilities" scope was used to pass parameters to the Cinder scheduler, options can be passed to the IBM Storwize/SVC driver with the "drivers" scope.

The following `extra_specs` keys are supported by the IBM Storwize/SVC driver:

- rsize
- warning
- autoexpand
- grainsize
- compression
- easytier
- multipath
- iogrp

These keys have the same semantics as their counterparts in the configuration file. They are set similarly; for example, `rsize=2` or `compression=False`.

Example: Volume types

In the following example, we create a volume type to specify a controller that supports iSCSI and compression, to use iSCSI when attaching the volume, and to enable compression:

```
$ cinder type-create compressed
$ cinder type-key compressed set capabilities:storage_protocol='<in> iSCSI'
capabilities:compression_support='<is> True' drivers:compression=True
```

We can then create a 50GB volume using this type:

```
$ cinder create --display-name "compressed volume" --volume-type compressed 50
```

Volume types can be used, for example, to provide users with different

- performance levels (such as, allocating entirely on an HDD tier, using Easy Tier for an HDD-SDD mix, or allocating entirely on an SSD tier)
- resiliency levels (such as, allocating volumes in pools with different RAID levels)
- features (such as, enabling/disabling Real-time Compression)

Operational notes for the Storwize family and SVC driver

Migrate volumes

In the context of OpenStack Block Storage's volume migration feature, the IBM Storwize/SVC driver enables the storage's virtualization technology. When migrating a volume from one pool to another, the volume will appear in the destination pool almost immediately, while the storage moves the data in the background.



Note

To enable this feature, both pools involved in a given volume migration must have the same values for `extent_size`. If the pools have different values for

`extent_size`, the data will still be moved directly between the pools (not host-side copy), but the operation will be synchronous.

Extend volumes

The IBM Storwize/SVC driver allows for extending a volume's size, but only for volumes without snapshots.

Snapshots and clones

Snapshots are implemented using FlashCopy with no background copy (space-efficient). Volume clones (volumes created from existing volumes) are implemented with FlashCopy, but with background copy enabled. This means that volume clones are independent, full copies. While this background copy is taking place, attempting to delete or extend the source volume will result in that operation waiting for the copy to complete.

Volume retype

The IBM Storwize/SVC driver enables you to modify volume types. When you modify volume types, you can also change these extra specs properties:

- `resize`
- `warning`
- `autoexpand`
- `grainsize`
- `compression`
- `easytier`
- `iogrp`



Note

When you change the `resize`, `grainsize` or `compression` properties, volume copies are asynchronously synchronized on the array.



Note

To change the `iogrp` property, IBM Storwize/SVC firmware version 6.4.0 or later is required.

LVM

The default volume back-end uses local volumes managed by LVM.

This driver supports different transport protocols to attach volumes, currently ISCSI and ISER.

Set the following in your `cinder.conf`, and use the following options to configure for ISCSI transport:

```
volume_driver = cinder.volume.drivers.lvm.LVMISCSIDriver
```

and for the iSER transport:

```
volume_driver = cinder.volume.drivers.lvm.LVMISERDriver
```

Table 1.13. Description of configuration options for lvm

Configuration option = Default value	Description
[DEFAULT]	
lvm_mirrors = 0	(IntOpt) If set, create lvms with multiple mirrors. Note that this requires lvm_mirrors + 2 pvs with available space
lvm_type = default	(StrOpt) Type of LVM volumes to deploy; (default or thin)
volume_group = cinder-volumes	(StrOpt) Name for the VG that will contain exported volumes

NetApp unified driver

The NetApp unified driver is a block storage driver that supports multiple storage families and protocols. A storage family corresponds to storage systems built on different NetApp technologies such as clustered Data ONTAP, Data ONTAP operating in 7-Mode, and E-Series. The storage protocol refers to the protocol used to initiate data storage and access operations on those storage systems like iSCSI and NFS. The NetApp unified driver can be configured to provision and manage OpenStack volumes on a given storage family using a specified storage protocol. The OpenStack volumes can then be used for accessing and storing data using the storage protocol on the storage family system. The NetApp unified driver is an extensible interface that can support new storage families and protocols.

NetApp clustered Data ONTAP storage family

The NetApp clustered Data ONTAP storage family represents a configuration group which provides OpenStack compute instances access to clustered Data ONTAP storage systems. At present it can be configured in OpenStack Block Storage to work with iSCSI and NFS storage protocols.

NetApp iSCSI configuration for clustered Data ONTAP

The NetApp iSCSI configuration for clustered Data ONTAP is an interface from OpenStack to clustered Data ONTAP storage systems for provisioning and managing the SAN block storage entity; that is, a NetApp LUN which can be accessed using the iSCSI protocol.

The iSCSI configuration for clustered Data ONTAP is a direct interface from OpenStack Block Storage to the clustered Data ONTAP instance and as such does not require additional management software to achieve the desired functionality. It uses NetApp APIs to interact with the clustered Data ONTAP instance.

Configuration options for clustered Data ONTAP family with iSCSI protocol

Configure the volume driver, storage family and storage protocol to the NetApp unified driver, clustered Data ONTAP, and iSCSI respectively by setting the `volume_driver`, `netapp_storage_family` and `netapp_storage_protocol` options in `cinder.conf` as follows:

```

volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family=ontap_cluster
netapp_storage_protocol=iscsi
netapp_vserver=openstack-vserver
netapp_server_hostname=myhostname
netapp_server_port=80
netapp_login=username
netapp_password=password

```



Note

You must override the default value of `netapp_storage_protocol` with `iscsi` in order to utilize the iSCSI protocol.

Table 1.14. Description of configuration options for `netapp_cdot_iscsi`

Configuration option = Default value	Description
[DEFAULT]	
<code>netapp_login = None</code>	(StrOpt) Administrative user account name used to access the storage system or proxy server.
<code>netapp_password = None</code>	(StrOpt) Password for the administrative user account specified in the <code>netapp_login</code> option.
<code>netapp_server_hostname = None</code>	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
<code>netapp_server_port = 80</code>	(IntOpt) The TCP port to use for communication with the storage system or proxy server. Traditionally, port 80 is used for HTTP and port 443 is used for HTTPS; however, this value should be changed if an alternate port has been configured on the storage system or proxy server.
<code>netapp_size_multiplier = 1.2</code>	(FloatOpt) The quantity to be multiplied by the requested volume size to ensure enough space is available on the virtual storage server (Vserver) to fulfill the volume creation request.
<code>netapp_storage_family = ontap_cluster</code>	(StrOpt) The storage family type used on the storage system; valid values are <code>ontap_7mode</code> for using Data ONTAP operating in 7-Mode, <code>ontap_cluster</code> for using clustered Data ONTAP, or <code>eseries</code> for using E-Series.
<code>netapp_storage_protocol = None</code>	(StrOpt) The storage protocol to be used on the data path with the storage system; valid values are <code>iscsi</code> or <code>nfs</code> .
<code>netapp_transport_type = http</code>	(StrOpt) The transport protocol used when communicating with the storage system or proxy server. Valid values are <code>http</code> or <code>https</code> .
<code>netapp_vserver = None</code>	(StrOpt) This option specifies the virtual storage server (Vserver) name on the storage cluster on which provisioning of block storage volumes should occur. If using the NFS storage protocol, this parameter is mandatory for storage service catalog support (utilized by Cinder volume type <code>extra_specs</code> support). If this option is specified, the exports belonging to the Vserver will only be used for provisioning in the future. Block storage volumes on exports not belonging to the Vserver specified by this option will continue to function normally.



Note

If you specify an account in the `netapp_login` that only has virtual storage server (Vserver) administration privileges (rather than cluster-wide administration privileges), some advanced features of the NetApp unified driver will not work and you may see warnings in the OpenStack Block Storage logs.



Tip

For more information on these options and other deployment and operational scenarios, visit the [OpenStack NetApp community](#).

NetApp NFS configuration for clustered Data ONTAP

The NetApp NFS configuration for clustered Data ONTAP is an interface from OpenStack to a clustered Data ONTAP system for provisioning and managing OpenStack volumes on NFS exports provided by the clustered Data ONTAP system that are accessed using the NFS protocol.

The NFS configuration for clustered Data ONTAP is a direct interface from OpenStack Block Storage to the clustered Data ONTAP instance and as such does not require any additional management software to achieve the desired functionality. It uses NetApp APIs to interact with the clustered Data ONTAP instance.

Configuration options for the clustered Data ONTAP family with NFS protocol

Configure the volume driver, storage family and storage protocol to NetApp unified driver, clustered Data ONTAP, and NFS respectively by setting the `volume_driver`, `netapp_storage_family` and `netapp_storage_protocol` options in `cinder.conf` as follows:

```
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family=ontap_cluster
netapp_storage_protocol=nfs
netapp_vserver=openstack-vserver
netapp_server_hostname=myhostname
netapp_server_port=80
netapp_login=username
netapp_password=password
nfs_shares_config=/etc/cinder/nfs_shares
```

Table 1.15. Description of configuration options for `netapp_cdot_nfs`

Configuration option = Default value	Description
[DEFAULT]	
<code>expiry_thres_minutes = 720</code>	(IntOpt) This option specifies the threshold for last access time for images in the NFS image cache. When a cache cleaning cycle begins, images in the cache that have not been accessed in the last M minutes, where M is the value of this parameter, will be deleted from the cache to create free space on the NFS share.
<code>netapp_copyoffload_tool_path = None</code>	(StrOpt) This option specifies the path of the NetApp copy offload tool binary. Ensure that the binary has execute permissions set which allow the effective user of the <code>cinder-volume</code> process to execute the file.

Configuration option = Default value	Description
netapp_login = None	(StrOpt) Administrative user account name used to access the storage system or proxy server.
netapp_password = None	(StrOpt) Password for the administrative user account specified in the netapp_login option.
netapp_server_hostname = None	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
netapp_server_port = 80	(IntOpt) The TCP port to use for communication with the storage system or proxy server. Traditionally, port 80 is used for HTTP and port 443 is used for HTTPS; however, this value should be changed if an alternate port has been configured on the storage system or proxy server.
netapp_storage_family = ontap_cluster	(StrOpt) The storage family type used on the storage system; valid values are ontap_7mode for using Data ONTAP operating in 7-Mode, ontap_cluster for using clustered Data ONTAP, or eseries for using E-Series.
netapp_storage_protocol = None	(StrOpt) The storage protocol to be used on the data path with the storage system; valid values are iscsi or nfs.
netapp_transport_type = http	(StrOpt) The transport protocol used when communicating with the storage system or proxy server. Valid values are http or https.
netapp_vserver = None	(StrOpt) This option specifies the virtual storage server (Vserver) name on the storage cluster on which provisioning of block storage volumes should occur. If using the NFS storage protocol, this parameter is mandatory for storage service catalog support (utilized by Cinder volume type extra_specs support). If this option is specified, the exports belonging to the Vserver will only be used for provisioning in the future. Block storage volumes on exports not belonging to the Vserver specified by this option will continue to function normally.
thres_avl_size_perc_start = 20	(IntOpt) If the percentage of available space for an NFS share has dropped below the value specified by this option, the NFS image cache will be cleaned.
thres_avl_size_perc_stop = 60	(IntOpt) When the percentage of available space on an NFS share has reached the percentage specified by this option, the driver will stop clearing files from the NFS image cache that have not been accessed in the last M minutes, where M is the value of the expiry_thres_minutes configuration option.



Note

Additional NetApp NFS configuration options are shared with the generic NFS driver. These options can be found here: [Table 1.22, "Description of configuration options for storage_nfs" \[63\]](#).



Note

If you specify an account in the `netapp_login` that only has virtual storage server (Vserver) administration privileges (rather than cluster-wide administration privileges), some advanced features of the NetApp unified driver will not work and you may see warnings in the OpenStack Block Storage logs.

NetApp NFS Copy Offload client

A feature was added in the Icehouse release of the NetApp unified driver that enables Image Service images to be efficiently copied to a destination Block Storage volume.

When the Block Storage and Image Service are configured to use the NetApp NFS Copy Offload client, a controller-side copy will be attempted before reverting to downloading the image from the Image Service. This improves image provisioning times while reducing the consumption of bandwidth and CPU cycles on the host(s) running the Image and Block Storage services. This is due to the copy operation being performed completely within the storage cluster.

The NetApp NFS Copy Offload client can be used in either of the following scenarios:

- The Image Service is configured to store images in an NFS share that is exported from a NetApp FlexVol volume *and* the destination for the new Block Storage volume will be on an NFS share exported from a different FlexVol volume than the one used by the Image Service. Both FlexVols must be located within the same cluster.
- The source image from the Image Service has already been cached in an NFS image cache within a Block Storage backend. The cached image resides on a different FlexVol volume than the destination for the new Block Storage volume. Both FlexVols must be located within the same cluster.

To use this feature, you must configure the Image Service, as follows:

- Set the `default_store` configuration option to `file`.
- Set the `filesystem_store_datadir` configuration option to the path to the Image Service NFS export.
- Set the `show_image_direct_url` configuration option to `True`.
- Set the `show_multiple_locations` configuration option to `True`.
- Set the `filesystem_store_metadata_file` configuration option to a metadata file. The metadata file should contain a JSON object that contains the correct information about the NFS export used by the Image Service, similar to:

```
{
  "share_location": "nfs://192.168.0.1/myGlanceExport",
  "mount_point": "/var/lib/glance/images",
  "type": "nfs"
}
```

To use this feature, you must configure the Block Storage service, as follows:

- Set the `netapp_copyoffload_tool_path` configuration option to the path to the NetApp Copy Offload binary.
- Set the `glance_api_version` configuration option to `2`.



Important

This feature requires that:

- The storage system must have Data ONTAP v8.2 or greater installed.
- The vStorage feature must be enabled on each storage virtual machine (SVM, also known as a Vserver) that is permitted to interact with the copy offload client.

- To configure the copy offload workflow, enable NFS v4.0 or greater and export it from the SVM.



Tip

To download the NetApp copy offload binary to be utilized in conjunction with the `netapp_copyoffload_tool_path` configuration option, please visit the Utility Toolchest page at the [NetApp Support portal](#) (login is required).



Tip

For more information on these options and other deployment and operational scenarios, visit the [OpenStack NetApp community](#).

NetApp-supported extra specs for clustered Data ONTAP

Extra specs enable vendors to specify extra filter criteria that the Block Storage scheduler uses when it determines which volume node should fulfill a volume provisioning request. When you use the NetApp unified driver with a clustered Data ONTAP storage system, you can leverage extra specs with OpenStack Block Storage volume types to ensure that OpenStack Block Storage volumes are created on storage back ends that have certain properties. For example, when you configure QoS, mirroring, or compression for a storage back end.

Extra specs are associated with OpenStack Block Storage volume types, so that when users request volumes of a particular volume type, the volumes are created on storage back ends that meet the list of requirements. For example, the back ends have the available space or extra specs. You can use the specs in the following table when you define OpenStack Block Storage volume types by using the `cinder type-key` command.

Table 1.16. Description of extra specs options for NetApp Unified Driver with Clustered Data ONTAP

Extra spec	Type	Description
<code>netapp:raid_type</code>	String	Limit the candidate volume list based on one of the following raid types: <code>raid4</code> , <code>raid_dp</code> .
<code>netapp:disk_type</code>	String	Limit the candidate volume list based on one of the following disk types: <code>ATA</code> , <code>BSAS</code> , <code>EATA</code> , <code>FCAL</code> , <code>FSAS</code> , <code>LUN</code> , <code>MSATA</code> , <code>SAS</code> , <code>SATA</code> , <code>SCSI</code> , <code>XATA</code> , <code>XSAS</code> , or <code>SSD</code> .
<code>netapp:qos_policy_group</code>	String	Specify the name of a QoS policy group, which defines measurable Service Level Objectives, that should be applied to the OpenStack Block Storage volume at the time of volume creation. Ensure that the QoS policy group object within Data ONTAP should be defined before an OpenStack Block Storage volume is created, and that the QoS policy group is not associated with the destination FlexVol volume.
<code>netapp_mirrored^a</code>	Boolean	Limit the candidate volume list to only the ones that are mirrored on the storage controller.
<code>netapp_unmirrored^a</code>	Boolean	Limit the candidate volume list to only the ones that are not mirrored on the storage controller.
<code>netapp_dedup^a</code>	Boolean	Limit the candidate volume list to only the ones that have deduplication enabled on the storage controller.
<code>netapp_nodedup^a</code>	Boolean	Limit the candidate volume list to only the ones that have deduplication disabled on the storage controller.

Extra spec	Type	Description
netapp_compression ^a	Boolean	Limit the candidate volume list to only the ones that have compression enabled on the storage controller.
netapp_nocompression ^a	Boolean	Limit the candidate volume list to only the ones that have compression disabled on the storage controller.
netapp_thin_provisioned ^a	Boolean	Limit the candidate volume list to only the ones that support thin provisioning on the storage controller.
netapp_thick_provisioned ^a	Boolean	Limit the candidate volume list to only the ones that support thick provisioning on the storage controller.

^aIf both the positive and negative specs for a pair are specified (for example, `netapp_dedup` and `netapp_nodedup`) and set to the same value within a single `extra_specs` list, then neither spec will be utilized by the driver.



Note

It is recommended to only set the value of extra specs to `True` when combining multiple specs to enforce a certain logic set. If you desire to remove volumes with a certain feature enabled from consideration from the OpenStack Block Storage volume scheduler, be sure to use the negated spec name with a value of `True` rather than setting the positive spec to a value of `False`.

NetApp Data ONTAP operating in 7-Mode storage family

The NetApp Data ONTAP operating in 7-Mode storage family represents a configuration group which provides OpenStack compute instances access to 7-Mode storage systems. At present it can be configured in OpenStack Block Storage to work with iSCSI and NFS storage protocols.

NetApp iSCSI configuration for Data ONTAP operating in 7-Mode

The NetApp iSCSI configuration for Data ONTAP operating in 7-Mode is an interface from OpenStack to Data ONTAP operating in 7-Mode storage systems for provisioning and managing the SAN block storage entity, that is, a LUN which can be accessed using iSCSI protocol.

The iSCSI configuration for Data ONTAP operating in 7-Mode is a direct interface from OpenStack to Data ONTAP operating in 7-Mode storage system and it does not require additional management software to achieve the desired functionality. It uses NetApp ONTAPI to interact with the Data ONTAP operating in 7-Mode storage system.

Configuration options for the Data ONTAP operating in 7-Mode storage family with iSCSI protocol

Configure the volume driver, storage family and storage protocol to the NetApp unified driver, Data ONTAP operating in 7-Mode, and iSCSI respectively by setting the `volume_driver`, `netapp_storage_family` and `netapp_storage_protocol` options in `cinder.conf` as follows:

```
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family=ontap_7mode
netapp_storage_protocol=iscsi
netapp_server_hostname=myhostname
netapp_server_port=80
netapp_login=username
netapp_password=password
```

**Note**

You must override the default value of `netapp_storage_protocol` with `iscsi` in order to utilize the iSCSI protocol.

Table 1.17. Description of configuration options for `netapp_7mode_iscsi`

Configuration option = Default value	Description
[DEFAULT]	
<code>netapp_login = None</code>	(StrOpt) Administrative user account name used to access the storage system or proxy server.
<code>netapp_password = None</code>	(StrOpt) Password for the administrative user account specified in the <code>netapp_login</code> option.
<code>netapp_server_hostname = None</code>	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
<code>netapp_server_port = 80</code>	(IntOpt) The TCP port to use for communication with the storage system or proxy server. Traditionally, port 80 is used for HTTP and port 443 is used for HTTPS; however, this value should be changed if an alternate port has been configured on the storage system or proxy server.
<code>netapp_size_multiplier = 1.2</code>	(FloatOpt) The quantity to be multiplied by the requested volume size to ensure enough space is available on the virtual storage server (Vserver) to fulfill the volume creation request.
<code>netapp_storage_family = ontap_cluster</code>	(StrOpt) The storage family type used on the storage system; valid values are <code>ontap_7mode</code> for using Data ONTAP operating in 7-Mode, <code>ontap_cluster</code> for using clustered Data ONTAP, or <code>eseries</code> for using E-Series.
<code>netapp_storage_protocol = None</code>	(StrOpt) The storage protocol to be used on the data path with the storage system; valid values are <code>iscsi</code> or <code>nfs</code> .
<code>netapp_transport_type = http</code>	(StrOpt) The transport protocol used when communicating with the storage system or proxy server. Valid values are <code>http</code> or <code>https</code> .
<code>netapp_vfiler = None</code>	(StrOpt) The vFiler unit on which provisioning of block storage volumes will be done. This option is only used by the driver when connecting to an instance with a storage family of Data ONTAP operating in 7-Mode and the storage protocol selected is iSCSI. Only use this option when utilizing the MultiStore feature on the NetApp storage system.
<code>netapp_volume_list = None</code>	(StrOpt) This option is only utilized when the storage protocol is configured to use iSCSI. This option is used to restrict provisioning to the specified controller volumes. Specify the value of this option to be a comma separated list of NetApp controller volume names to be used for provisioning.

**Tip**

For more information on these options and other deployment and operational scenarios, visit the [OpenStack NetApp community](#).

NetApp NFS configuration for Data ONTAP operating in 7-Mode

The NetApp NFS configuration for Data ONTAP operating in 7-Mode is an interface from OpenStack to Data ONTAP operating in 7-Mode storage system for provisioning and

managing OpenStack volumes on NFS exports provided by the Data ONTAP operating in 7-Mode storage system which can then be accessed using NFS protocol.

The NFS configuration for Data ONTAP operating in 7-Mode is a direct interface from OpenStack Block Storage to the Data ONTAP operating in 7-Mode instance and as such does not require any additional management software to achieve the desired functionality. It uses NetApp ONTAPI to interact with the Data ONTAP operating in 7-Mode storage system.

Configuration options for the Data ONTAP operating in 7-Mode family with NFS protocol

Configure the volume driver, storage family and storage protocol to the NetApp unified driver, Data ONTAP operating in 7-Mode, and NFS respectively by setting the `volume_driver`, `netapp_storage_family` and `netapp_storage_protocol` options in `cinder.conf` as follows:

```
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family=ontap_7mode
netapp_storage_protocol=nfs
netapp_server_hostname=myhostname
netapp_server_port=80
netapp_login=username
netapp_password=password
nfs_shares_config=/etc/cinder/nfs_shares
```

Table 1.18. Description of configuration options for netapp_7mode_nfs

Configuration option = Default value	Description
[DEFAULT]	
<code>expiry_thres_minutes = 720</code>	(IntOpt) This option specifies the threshold for last access time for images in the NFS image cache. When a cache cleaning cycle begins, images in the cache that have not been accessed in the last M minutes, where M is the value of this parameter, will be deleted from the cache to create free space on the NFS share.
<code>netapp_login = None</code>	(StrOpt) Administrative user account name used to access the storage system or proxy server.
<code>netapp_password = None</code>	(StrOpt) Password for the administrative user account specified in the <code>netapp_login</code> option.
<code>netapp_server_hostname = None</code>	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
<code>netapp_server_port = 80</code>	(IntOpt) The TCP port to use for communication with the storage system or proxy server. Traditionally, port 80 is used for HTTP and port 443 is used for HTTPS; however, this value should be changed if an alternate port has been configured on the storage system or proxy server.
<code>netapp_storage_family = ontap_cluster</code>	(StrOpt) The storage family type used on the storage system; valid values are <code>ontap_7mode</code> for using Data ONTAP operating in 7-Mode, <code>ontap_cluster</code> for using clustered Data ONTAP, or <code>eseries</code> for using E-Series.
<code>netapp_storage_protocol = None</code>	(StrOpt) The storage protocol to be used on the data path with the storage system; valid values are <code>iscsi</code> or <code>nfs</code> .
<code>netapp_transport_type = http</code>	(StrOpt) The transport protocol used when communicating with the storage system or proxy server. Valid values are <code>http</code> or <code>https</code> .

Configuration option = Default value	Description
thres_avl_size_perc_start = 20	(IntOpt) If the percentage of available space for an NFS share has dropped below the value specified by this option, the NFS image cache will be cleaned.
thres_avl_size_perc_stop = 60	(IntOpt) When the percentage of available space on an NFS share has reached the percentage specified by this option, the driver will stop clearing files from the NFS image cache that have not been accessed in the last M minutes, where M is the value of the expiry_thres_minutes configuration option.



Note

Additional NetApp NFS configuration options are shared with the generic NFS driver. These options can be found here: [Table 1.22, "Description of configuration options for storage_nfs" \[63\]](#).



Tip

For more information on these options and other deployment and operational scenarios, visit the [OpenStack NetApp community](#).

NetApp E-Series storage family

The NetApp E-Series storage family represents a configuration group which provides OpenStack compute instances access to E-Series storage systems. At present it can be configured in OpenStack Block Storage to work with the iSCSI storage protocol.

NetApp iSCSI configuration for E-Series

The NetApp iSCSI configuration for E-Series is an interface from OpenStack to E-Series storage systems for provisioning and managing the SAN block storage entity; that is, a NetApp LUN which can be accessed using the iSCSI protocol.

The iSCSI configuration for E-Series is an interface from OpenStack Block Storage to the E-Series proxy instance and as such requires the deployment of the proxy instance in order to achieve the desired functionality. The driver uses REST APIs to interact with the E-Series proxy instance, which in turn interacts directly with the E-Series controllers.

Configuration options for E-Series storage family with iSCSI protocol

Configure the volume driver, storage family and storage protocol to the NetApp unified driver, E-Series, and iSCSI respectively by setting the `volume_driver`, `netapp_storage_family` and `netapp_storage_protocol` options in `cinder.conf` as follows:

```

volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family=eseries
netapp_storage_protocol=iscsi
netapp_server_hostname=myhostname
netapp_server_port=80
netapp_login=username
netapp_password=password
netapp_controller_ips=1.2.3.4,5.6.7.8
netapp_sa_password=arrayPassword
netapp_storage_pools=pool1,pool2

```

**Note**

You must override the default value of `netapp_storage_family` with `eseries` in order to utilize the E-Series driver.

**Note**

You must override the default value of `netapp_storage_protocol` with `iscsi` in order to utilize the iSCSI protocol.

Table 1.19. Description of configuration options for `netapp_eseries_iscsi`

Configuration option = Default value	Description
[DEFAULT]	
<code>netapp_controller_ips = None</code>	(StrOpt) This option is only utilized when the storage family is configured to <code>eseries</code> . This option is used to restrict provisioning to the specified controllers. Specify the value of this option to be a comma separated list of controller hostnames or IP addresses to be used for provisioning.
<code>netapp_eseries_host_type = linux_dm_mp</code>	(StrOpt) This option is used to define how the controllers in the E-Series storage array will work with the particular operating system on the hosts that are connected to it.
<code>netapp_login = None</code>	(StrOpt) Administrative user account name used to access the storage system or proxy server.
<code>netapp_password = None</code>	(StrOpt) Password for the administrative user account specified in the <code>netapp_login</code> option.
<code>netapp_sa_password = None</code>	(StrOpt) Password for the NetApp E-Series storage array.
<code>netapp_server_hostname = None</code>	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
<code>netapp_server_port = 80</code>	(IntOpt) The TCP port to use for communication with the storage system or proxy server. Traditionally, port 80 is used for HTTP and port 443 is used for HTTPS; however, this value should be changed if an alternate port has been configured on the storage system or proxy server.
<code>netapp_storage_family = ontap_cluster</code>	(StrOpt) The storage family type used on the storage system; valid values are <code>ontap_7mode</code> for using Data ONTAP operating in 7-Mode, <code>ontap_cluster</code> for using clustered Data ONTAP, or <code>eseries</code> for using E-Series.
<code>netapp_storage_pools = None</code>	(StrOpt) This option is used to restrict provisioning to the specified storage pools. Only dynamic disk pools are currently supported. Specify the value of this option to be a comma separated list of disk pool names to be used for provisioning.

Configuration option = Default value	Description
netapp_transport_type = http	(StrOpt) The transport protocol used when communicating with the storage system or proxy server. Valid values are http or https.
netapp_webservice_path = /devmgr/v2	(StrOpt) This option is used to specify the path to the E-Series proxy application on a proxy server. The value is combined with the value of the netapp_transport_type, netapp_server_hostname, and netapp_server_port options to create the URL used by the driver to connect to the proxy application.



Tip

For more information on these options and other deployment and operational scenarios, visit the [OpenStack NetApp community](#).

Upgrading prior NetApp drivers to the NetApp unified driver

NetApp introduced a new unified block storage driver in Havana for configuring different storage families and storage protocols. This requires defining upgrade path for NetApp drivers which existed in releases prior to Havana. This section covers the upgrade configuration for NetApp drivers to the new unified configuration and a list of deprecated NetApp drivers.

Upgraded NetApp drivers

This section describes how to update OpenStack Block Storage configuration from a pre-Havana release to the new unified driver format.

Driver upgrade configuration

1. NetApp iSCSI direct driver for Clustered Data ONTAP in Grizzly (or earlier)

```
volume_driver=cinder.volume.drivers.netapp.iscsi.  
NetAppDirectCmodeISCSIDriver
```

NetApp Unified Driver configuration

```
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver  
netapp_storage_family=ontap_cluster  
netapp_storage_protocol=iscsi
```

2. NetApp NFS direct driver for Clustered Data ONTAP in Grizzly (or earlier)

```
volume_driver=cinder.volume.drivers.netapp.nfs.NetAppDirectCmodeNfsDriver
```

NetApp Unified Driver configuration


```
volume_driver=cinder.volume.drivers.netapp.nfs.NetAppNFSDriver
```



Note

See the [OpenStack NetApp community](#) for support information on deprecated NetApp drivers in the Havana release.

Nexenta drivers

NexentaStor Appliance is NAS/SAN software platform designed for building reliable and fast network storage arrays. The Nexenta Storage Appliance uses ZFS as a disk management system. NexentaStor can serve as a storage node for the OpenStack and its virtual servers through iSCSI and NFS protocols.

With the NFS option, every Compute volume is represented by a directory designated to be its own file system in the ZFS file system. These file systems are exported using NFS.

With either option some minimal setup is required to tell OpenStack which NexentaStor servers are being used, whether they are supporting iSCSI and/or NFS and how to access each of the servers.

Typically the only operation required on the NexentaStor servers is to create the containing directory for the iSCSI or NFS exports. For NFS this containing directory must be explicitly exported via NFS. There is no software that must be installed on the NexentaStor servers; they are controlled using existing management plane interfaces.

Nexenta iSCSI driver

The Nexenta iSCSI driver allows you to use a NexentaStor appliance to store Compute volumes. Every Compute volume is represented by a single zvol in a predefined Nexenta namespace. For every new volume the driver creates a iSCSI target and iSCSI target group that are used to access it from compute hosts.

The Nexenta iSCSI volume driver should work with all versions of NexentaStor. The NexentaStor appliance must be installed and configured according to the relevant Nexenta documentation. A pool and an enclosing namespace must be created for all iSCSI volumes to be accessed through the volume driver. This should be done as specified in the release specific NexentaStor documentation.

The NexentaStor Appliance iSCSI driver is selected using the normal procedures for one or multiple back-end volume drivers. You must configure these items for each NexentaStor appliance that the iSCSI volume driver controls:

Enable the Nexenta iSCSI driver and related options

This table contains the options supported by the Nexenta iSCSI driver.

Table 1.20. Description of configuration options for nexenta_iscsi

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
nexenta_blocksize =	(StrOpt) block size for volumes (blank=default,8KB)
nexenta_host =	(StrOpt) IP address of Nexenta SA
nexenta_iscsi_target_portal_port = 3260	(IntOpt) Nexenta target portal port
nexenta_password = nexenta	(StrOpt) Password to connect to Nexenta SA
nexenta_rest_port = 2000	(IntOpt) HTTP port to connect to Nexenta REST API server
nexenta_rest_protocol = auto	(StrOpt) Use http or https for REST connection (default auto)
nexenta_rrmgr_compression = 0	(IntOpt) Enable stream compression, level 1..9. 1 - gives best speed; 9 - gives best compression.
nexenta_rrmgr_connections = 2	(IntOpt) Number of TCP connections.
nexenta_rrmgr_tcp_buf_size = 4096	(IntOpt) TCP Buffer size in KiloBytes.
nexenta_sparse = False	(BoolOpt) flag to create sparse volumes
nexenta_sparsed_volumes = True	(BoolOpt) Create volumes as sparsed files which take no space.If set to False volume is created as regular file.In such case volume creation takes a lot of time.
nexenta_target_group_prefix = cinder/	(StrOpt) prefix for iSCSI target groups on SA
nexenta_target_prefix = iqn.1986-03.com.sun:02:cinder-	(StrOpt) IQN prefix for iSCSI targets
nexenta_user = admin	(StrOpt) User name to connect to Nexenta SA
nexenta_volume = cinder	(StrOpt) pool on SA that will hold all volumes

To use Compute with the Nexenta iSCSI driver, first set the `volume_driver`:

```
volume_driver=cinder.volume.drivers.nexenta.iscsi.NexentaISCSIDriver
```

Then, set the `nexenta_host` parameter and other parameters from the table, if needed.

Nexenta NFS driver

The Nexenta NFS driver allows you to use NexentaStor appliance to store Compute volumes via NFS. Every Compute volume is represented by a single NFS file within a shared directory.

While the NFS protocols standardize file access for users, they do not standardize administrative actions such as taking snapshots or replicating file systems. The OpenStack Volume Drivers bring a common interface to these operations. The Nexenta NFS driver implements these standard actions using the ZFS management plane that already is deployed on NexentaStor appliances.

The Nexenta NFS volume driver should work with all versions of NexentaStor. The NexentaStor appliance must be installed and configured according to the relevant Nexenta documentation. A single-parent file system must be created for all virtual disk directories supported for OpenStack. This directory must be created and exported on each NexentaStor appliance. This should be done as specified in the release specific NexentaStor documentation.

Enable the Nexenta NFS driver and related options

To use Compute with the Nexenta NFS driver, first set the `volume_driver`:

```
volume_driver = cinder.volume.drivers.nexenta.nfs.NexentaNfsDriver
```

The following table contains the options supported by the Nexenta NFS driver.

Table 1.21. Description of configuration options for nexenta_nfs

Configuration option = Default value	Description
[DEFAULT]	
nexenta_mount_point_base = \$state_path/mnt	(StrOpt) Base dir containing mount points for nfs shares
nexenta_nms_cache_volroot = True	(BoolOpt) If set True cache NexentaStor appliance volroot option value.
nexenta_shares_config = /etc/cinder/nfs_shares	(StrOpt) File with the list of available nfs shares
nexenta_volume_compression = on	(StrOpt) Default compression value for new ZFS folders.

Add your list of Nexenta NFS servers to the file you specified with the `nexenta_shares_config` option. For example, if the value of this option was set to `/etc/cinder/nfs_shares`, then:

```
# cat /etc/cinder/nfs_shares
192.168.1.200:/storage http://admin:nexenta@192.168.1.200:2000
192.168.1.201:/storage http://admin:nexenta@192.168.1.201:2000
192.168.1.202:/storage http://admin:nexenta@192.168.1.202:2000
```

Comments are allowed in this file. They begin with a #.

Each line in this file represents a NFS share. The first part of the line is the NFS share URL, the second is the connection URL to the NexentaStor Appliance.

NFS driver

The Network File System (NFS) is a distributed file system protocol originally developed by Sun Microsystems in 1984. An NFS server *exports* one or more of its file systems, known as *shares*. An NFS client can mount these exported shares on its own file system. You can perform file actions on this mounted remote file system as if the file system were local.

How the NFS driver works

The NFS driver, and other drivers based on it, work quite differently than a traditional block storage driver.

The NFS driver does not actually allow an instance to access a storage device at the block level. Instead, files are created on an NFS share and mapped to instances, which emulates a block device. This works in a similar way to QEMU, which stores instances in the `/var/lib/nova/instances` directory.

Enable the NFS driver and related options

To use Cinder with the NFS driver, first set the `volume_driver` in `cinder.conf`:

```
volume_driver=cinder.volume.drivers.nfs.NfsDriver
```

The following table contains the options supported by the NFS driver.

Table 1.22. Description of configuration options for storage_nfs

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
<code>nfs_mount_options = None</code>	(StrOpt) Mount options passed to the nfs client. See section of the nfs man page for details.
<code>nfs_mount_point_base = \$state_path/mnt</code>	(StrOpt) Base dir containing mount points for nfs shares.
<code>nfs_oversub_ratio = 1.0</code>	(FloatOpt) This will compare the allocated to available space on the volume destination. If the ratio exceeds this number, the destination will no longer be valid.
<code>nfs_shares_config = /etc/cinder/nfs_shares</code>	(StrOpt) File with the list of available nfs shares
<code>nfs_sparsed_volumes = True</code>	(BoolOpt) Create volumes as sparsed files which take no space. If set to False volume is created as regular file. In such case volume creation takes a lot of time.
<code>nfs_used_ratio = 0.95</code>	(FloatOpt) Percent of ACTUAL usage of the underlying volume before no new volumes can be allocated to the volume destination.



Note

As of the Icehouse release, the NFS driver (and other drivers based off it) will attempt to mount shares using version 4.1 of the NFS protocol (including pNFS). If the mount attempt is unsuccessful due to a lack of client or server support, a subsequent mount attempt that requests the default behavior of the `mount.nfs` command will be performed. On most distributions, the default behavior is to attempt mounting first with NFS v4.0, then silently fall back to NFS v3.0 if necessary. If the `nfs_mount_options` configuration option contains a request for a specific version of NFS to be used, or if specific options are specified in the shares configuration file specified by the `nfs_shares_config` configuration option, the mount will be attempted as requested with no subsequent attempts.

How to use the NFS driver

1. Access to one or more NFS servers. Creating an NFS server is outside the scope of this document. This example assumes access to the following NFS servers and mount points:

- `192.168.1.200:/storage`
- `192.168.1.201:/storage`
- `192.168.1.202:/storage`

This example demonstrates the use of with this driver with multiple NFS servers. Multiple servers are not required. One is usually enough.

2. Add your list of NFS servers to the file you specified with the `nfs_shares_config` option. For example, if the value of this option was set to `/etc/cinder/shares.txt`, then:

```
# cat /etc/cinder/shares.txt
192.168.1.200:/storage
192.168.1.201:/storage
192.168.1.202:/storage
```

Comments are allowed in this file. They begin with a #.

3. Configure the `nfs_mount_point_base` option. This is a directory where `cinder-volume` mounts all NFS shares stored in `shares.txt`. For this example, `/var/lib/cinder/nfs` is used. You can, of course, use the default value of `$state_path/mnt`.
4. Start the `cinder-volume` service. `/var/lib/cinder/nfs` should now contain a directory for each NFS share specified in `shares.txt`. The name of each directory is a hashed name:

```
# ls /var/lib/cinder/nfs/  
...  
46c5db75dc3a3a50a10bfd1a456a9f3f  
...
```

5. You can now create volumes as you normally would:

```
$ nova volume-create --display-name=myvol 5  
# ls /var/lib/cinder/nfs/46c5db75dc3a3a50a10bfd1a456a9f3f  
volume-a8862558-e6d6-4648-b5df-bb84f31c8935
```

This volume can also be attached and deleted just like other volumes. However, snapshotting is *not* supported.

NFS driver notes

- `cinder-volume` manages the mounting of the NFS shares as well as volume creation on the shares. Keep this in mind when planning your OpenStack architecture. If you have one master NFS server, it might make sense to only have one `cinder-volume` service to handle all requests to that NFS server. However, if that single server is unable to handle all requests, more than one `cinder-volume` service is needed as well as potentially more than one NFS server.
- Because data is stored in a file and not actually on a block storage device, you might not see the same IO performance as you would with a traditional block storage driver. Please test accordingly.
- Despite possible IO performance loss, having volume data stored in a file might be beneficial. For example, backing up volumes can be as easy as copying the volume files.



Note

Regular IO flushing and syncing still stands.

SolidFire

The SolidFire Cluster is a high performance all SSD iSCSI storage device that provides massive scale out capability and extreme fault tolerance. A key feature of the SolidFire cluster is the ability to set and modify during operation specific QoS levels on a volume for volume basis. The SolidFire cluster offers this along with de-duplication, compression, and an architecture that takes full advantage of SSDs.

To configure the use of a SolidFire cluster with Block Storage, modify your `cinder.conf` file as follows:

```

volume_driver = cinder.volume.drivers.solidfire.SolidFireDriver
san_ip = 172.17.1.182           # the address of your MVIP
san_login = sfadmin            # your cluster admin login
san_password = sfpasword      # your cluster admin password
sf_account_prefix = ''        # prefix for tenant account creation on
                               solidfire cluster (see warning below)

```



Warning

The SolidFire driver creates a unique account prefixed with `$cinder-volume-service-hostname-$tenant-id` on the SolidFire cluster for each tenant that accesses the cluster through the Volume API. Unfortunately, this account formation results in issues for High Availability (HA) installations and installations where the `cinder-volume` service can move to a new node. HA installations can return an Account Not Found error because the call to the SolidFire cluster is not always going to be sent from the same node. In installations where the `cinder-volume` service moves to a new node, the same issue can occur when you perform operations on existing volumes, such as clone, extend, delete, and so on.



Note

Set the `sf_account_prefix` option to an empty string (") in the `cinder.conf` file. This setting results in unique accounts being created on the SolidFire cluster, but the accounts are prefixed with the `tenant-id` or any unique identifier that you choose and are independent of the host where the `cinder-volume` service resides.

Table 1.23. Description of configuration options for solidfire

Configuration option = Default value	Description
[DEFAULT]	
<code>sf_account_prefix = None</code>	(StrOpt) Create SolidFire accounts with this prefix. Any string can be used here, but the string "hostname" is special and will create a prefix using the cinder node hostname (previous default behavior). The default is NO prefix.
<code>sf_allow_tenant_qos = False</code>	(BoolOpt) Allow tenants to specify QOS on create
<code>sf_api_port = 443</code>	(IntOpt) SolidFire API port. Useful if the device api is behind a proxy on a different port.
<code>sf_emulate_512 = True</code>	(BoolOpt) Set 512 byte emulation on volume creation;

VMware VMDK driver

Use the VMware VMDK driver to enable management of the OpenStack Block Storage volumes on vCenter-managed data stores. Volumes are backed by VMDK files on data stores that use any VMware-compatible storage technology such as NFS, iSCSI, FiberChannel, and vSAN.



Warning

The VMware ESX VMDK driver is deprecated as of the Icehouse release and might be removed in Juno or a subsequent release. The VMware vCenter VMDK driver continues to be fully supported.

Functional context

The VMware VMDK driver connects to vCenter, through which it can dynamically access all the data stores visible from the ESX hosts in the managed cluster.

When you create a volume, the VMDK driver creates a VMDK file on demand. The VMDK file creation completes only when the volume is subsequently attached to an instance, because the set of data stores visible to the instance determines where to place the volume.

The running vSphere VM is automatically reconfigured to attach the VMDK file as an extra disk. Once attached, you can log in to the running vSphere VM to rescan and discover this extra disk.

Configuration

The recommended volume driver for OpenStack Block Storage is the VMware vCenter VMDK driver. When you configure the driver, you must match it with the appropriate OpenStack Compute driver from VMware and both drivers must point to the same server.

In the `nova.conf` file, use this option to define the Compute driver:

```
compute_driver=vmwareapi.VMwareVCDriver
```

In the `cinder.conf` file, use this option to define the volume driver:

```
volume_driver=cinder.volume.drivers.vmware.vmdk.VMwareVcVmdkDriver
```

The following table lists various options that the drivers support for the OpenStack Block Storage configuration (`cinder.conf`):

Table 1.24. Description of configuration options for vmware

Configuration option = Default value	Description
[DEFAULT]	
vmware_api_retry_count = 10	(IntOpt) Number of times VMware ESX/VC server API must be retried upon connection related issues.
vmware_host_ip = None	(StrOpt) IP address for connecting to VMware ESX/VC server.
vmware_host_password = None	(StrOpt) Password for authenticating with VMware ESX/VC server.
vmware_host_username = None	(StrOpt) Username for authenticating with VMware ESX/VC server.
vmware_host_version = None	(StrOpt) Optional string specifying the VMware VC server version. The driver attempts to retrieve the version from VMware VC server. Set this configuration only if you want to override the VC server version.
vmware_image_transfer_timeout_secs = 7200	(IntOpt) Timeout in seconds for VMDK volume transfer between Cinder and Glance.
vmware_max_objects_retrieval = 100	(IntOpt) Max number of objects to be retrieved per batch. Query results will be obtained in batches from the server and not in one shot. Server may still limit the count to something less than the configured value.
vmware_task_poll_interval = 5	(IntOpt) The interval (in seconds) for polling remote tasks invoked on VMware ESX/VC server.
vmware_volume_folder = cinder-volumes	(StrOpt) Name for the folder in the VC datacenter that will contain cinder volumes.

Configuration option = Default value	Description
vmware_wsdl_location = None	(StrOpt) Optional VIM service WSDL Location e.g http://<server>/vimService.wsdl. Optional over-ride to default location for bug work-arounds.

VMDK disk type

The VMware VMDK drivers support the creation of VMDK disk files of type `thin`, `thick`, or `eagerZeroedThick`. Use the `vmware:vmdk_type` extra spec key with the appropriate value to specify the VMDK disk file type. The following table captures the mapping between the extra spec entry and the VMDK disk file type:

Table 1.25. Extra spec entry to VMDK disk file type mapping

Disk file type	Extra spec key	Extra spec value
thin	vmware:vmdk_type	thin
thick	vmware:vmdk_type	thick
eagerZeroedThick	vmware:vmdk_type	eagerZeroedThick

If you do not specify a `vmdk_type` extra spec entry, the default disk file type is `thin`.

The following example shows how to create a `thick` VMDK volume by using the appropriate `vmdk_type`:

```
$ cinder type-create thick_volume
$ cinder type-key thick_volume set vmware:vmdk_type=thick
$ cinder create --volume-type thick_volume --display-name volume1 1
```

Clone type

With the VMware VMDK drivers, you can create a volume from another source volume or a snapshot point. The VMware vCenter VMDK driver supports the `full` and `linked/fast` clone types. Use the `vmware:clone_type` extra spec key to specify the clone type. The following table captures the mapping for clone types:

Table 1.26. Extra spec entry to clone type mapping

Clone type	Extra spec key	Extra spec value
full	vmware:clone_type	full
linked/fast	vmware:clone_type	linked

If you do not specify the clone type, the default is `full`.

The following example shows linked cloning from another source volume:

```
$ cinder type-create fast_clone
$ cinder type-key fast_clone set vmware:clone_type=linked
$ cinder create --volume-type fast_clone --source-volid 25743b9d-3605-462b-
b9eb-71459fe2bb35 --display-name volume1 1
```



Note

The VMware ESX VMDK driver ignores the extra spec entry and always creates a `full` clone.

Use vCenter storage policies to specify back-end data stores

This section describes how to configure back-end data stores using storage policies. In vCenter, you can create one or more storage policies and expose them as a Block Storage volume-type to a vmdk volume. The storage policies are exposed to the vmdk driver through the extra spec property with the `vmware:storage_profile` key.

For example, assume a storage policy in vCenter named `gold_policy`. and a Block Storage volume type named `vol1` with the extra spec key `vmware:storage_profile` set to the value `gold_policy`. Any Block Storage volume creation that uses the `vol1` volume type places the volume only in data stores that match the `gold_policy` storage policy.

The Block Storage back-end configuration for vSphere data stores is automatically determined based on the vCenter configuration. If you configure a connection to connect to vCenter version 5.5 or later in the `cinder.conf` file, the use of storage policies to configure back-end data stores is automatically supported.



Note

You must configure any data stores that you configure for the Block Storage service for the Compute service.

Procedure 1.6. To configure back-end data stores by using storage policies

1. In vCenter, tag the data stores to be used for the back end.

OpenStack also supports policies that are created by using vendor-specific capabilities; for example vSAN-specific storage policies.



Note

The tag value serves as the policy. For details, see [the section called "Storage policy-based configuration in vCenter" \[70\]](#).

2. Set the extra spec key `vmware:storage_profile` in the desired Block Storage volume types to the policy name that you created in the previous step.
3. Optionally, for the `vmware_host_version` parameter, enter the version number of your vSphere platform. For example, `5.5`.

This setting overrides the default location for the corresponding WSDL file. Among other scenarios, you can use this setting to prevent WSDL error messages during the development phase or to work with a newer version of vCenter.

4. Complete the other vCenter configuration parameters as appropriate.



Note

The following considerations apply to configuring SPBM for the Block Storage service:

- Any volume that is created without an associated policy (that is to say, without an associated volume type that specifies `vmware:storage_profile` extra spec), there is no policy-based placement for that volume.

Supported operations

The VMware vCenter and ESX VMDK drivers support these operations:

- Create volume
- Create volume from another source volume. (Supported only if source volume is not attached to an instance.)
- Create volume from snapshot
- Create volume from glance image
- Attach volume (When a volume is attached to an instance, a reconfigure operation is performed on the instance to add the volume's VMDK to it. The user must manually rescan and mount the device from within the guest operating system.)
- Detach volume
- Create snapshot (Allowed only if volume is not attached to an instance.)
- Delete snapshot (Allowed only if volume is not attached to an instance.)
- Upload as image to glance (Allowed only if volume is not attached to an instance.)



Note

Although the VMware ESX VMDK driver supports these operations, it has not been extensively tested.

Storage policy-based configuration in vCenter

You can configure Storage Policy-Based Management (SPBM) profiles for vCenter data stores supporting the Compute, Image Service, and Block Storage components of an OpenStack implementation.

In a vSphere OpenStack deployment, SPBM enables you to delegate several data stores for storage, which reduces the risk of running out of storage space. The policy logic selects the data store based on accessibility and available storage space.

Prerequisites

- Determine the data stores to be used by the SPBM policy.
- Determine the tag that identifies the data stores in the OpenStack component configuration.

- Create separate policies or sets of data stores for separate OpenStack components.

Create storage policies in vCenter

Procedure 1.7. To create storage policies in vCenter

1. In vCenter, create the tag that identifies the data stores:
 - a. From the Home screen, click **Tags**.
 - b. Specify a name for the tag.
 - c. Specify a tag category. For example, `spbm-cinder`.
2. Apply the tag to the data stores to be used by the SPBM policy.



Note

For details about creating tags in vSphere, see the [vSphere documentation](#).

3. In vCenter, create a tag-based storage policy that uses one or more tags to identify a set of data stores.



Note

You use this tag name and category when you configure the `*.conf` file for the OpenStack component. For details about creating tags in vSphere, see the [vSphere documentation](#).

Data store selection

If storage policy is enabled, the driver initially selects all the data stores that match the associated storage policy.

If two or more data stores match the storage policy, the driver chooses a data store that is connected to the maximum number of hosts.

In case of ties, the driver chooses the data store with lowest space utilization, where space utilization is defined by the $(1 - \text{freespace} / \text{totalspace})$ metric.

These actions reduce the number of volume migrations while attaching the volume to instances.

The volume must be migrated if the ESX host for the instance cannot access the data store that contains the volume.

Windows

There is a volume back-end for Windows. Set the following in your `cinder.conf`, and use the options below to configure it.

```
volume_driver=cinder.volume.drivers.windows.WindowsDriver
```

Table 1.27. Description of configuration options for windows

Configuration option = Default value	Description
[DEFAULT]	
windows_iscsi_lun_path = C:\jSCSIVirtualDisks	(StrOpt) Path to store VHD backed volumes

XenAPI NFS

XenAPI NFS is a Block Storage (Cinder) driver that uses an NFS share through the XenAPI Storage Manager to store virtual disk images and expose those virtual disks as volumes.

This driver does not access the NFS share directly. It accesses the share only through XenAPI Storage Manager. Consider this driver as a reference implementation for use of the XenAPI Storage Manager in OpenStack (present in XenServer and XCP).

Requirements

- A XenServer/XCP installation that acts as Storage Controller. This hypervisor is known as the storage controller.
- Use XenServer/XCP as your hypervisor for Compute nodes.
- An NFS share that is configured for XenServer/XCP. For specific requirements and export options, see the administration guide for your specific XenServer version. The NFS share must be accessible by all XenServers components within your cloud.
- To create volumes from XenServer type images (vhd tgz files), XenServer Nova plug-ins are also required on the storage controller.

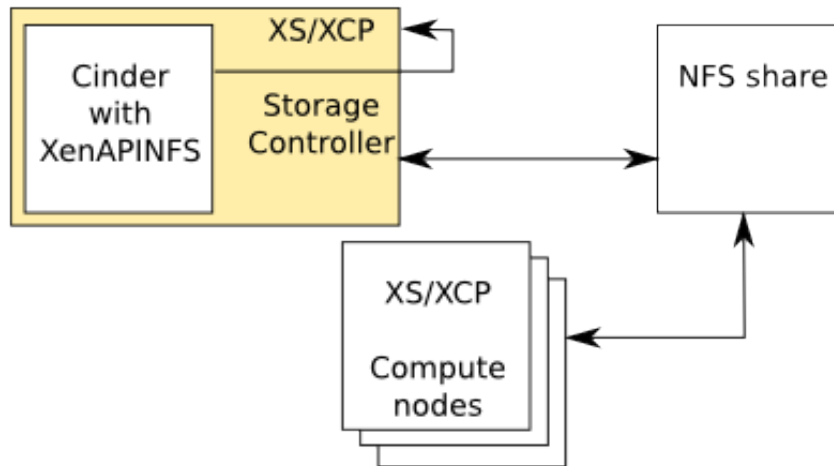


Note

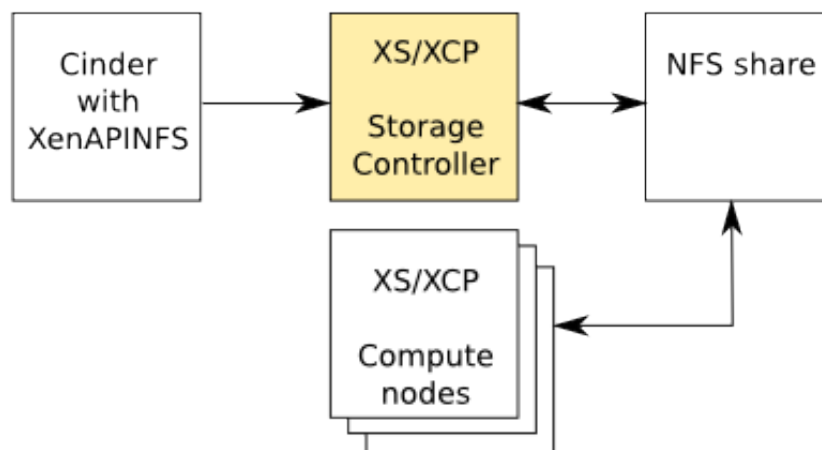
You can use a XenServer as a storage controller and compute node at the same time. This minimal configuration consists of a XenServer/XCP box and an NFS share.

Configuration patterns

- Local configuration (Recommended): The driver runs in a virtual machine on top of the storage controller. With this configuration, you can create volumes from `qemu-img`-supported formats.

Figure 1.3. Local configuration

- Remote configuration: The driver is not a guest VM of the storage controller. With this configuration, you can only use XenServer vhd-type images to create volumes.

Figure 1.4. Remote configuration

Configuration options

Assuming the following setup:

- XenServer box at 10.2.2.1
- XenServer password is r00tme
- NFS server is nfs.example.com
- NFS export is at /volumes

To use XenAPI/NFS as your cinder driver, set these configuration options in the `cinder.conf` file:

```

volume_driver = cinder.volume.drivers.xenapi.sm.XenAPIINFSDriver
xenapi_connection_url = http://10.2.2.1
xenapi_connection_username = root
xenapi_connection_password = r00tme
xenapi_nfs_server = nfs.example.com
xenapi_nfs_serverpath = /volumes

```

The following table shows the configuration options that the XenAPIINFS driver supports:

Table 1.28. Description of configuration options for storage_xen

Configuration option = Default value	Description
[DEFAULT]	
xenapi_connection_password = None	(StrOpt) Password for XenAPI connection
xenapi_connection_url = None	(StrOpt) URL for XenAPI connection
xenapi_connection_username = root	(StrOpt) Username for XenAPI connection
xenapi_nfs_server = None	(StrOpt) NFS server to be used by XenAPIINFSDriver
xenapi_nfs_serverpath = None	(StrOpt) Path of exported NFS, used by XenAPIINFSDriver
xenapi_sr_base_path = /var/run/sr-mount	(StrOpt) Base path to the storage repository

XenAPI Storage Manager volume driver

The Xen Storage Manager volume driver (xensm) is a XenAPI hypervisor specific volume driver, and can be used to provide basic storage functionality, including volume creation and destruction, on a number of different storage back-ends. It also enables the capability of using more sophisticated storage back-ends for operations like cloning/snapshots, and so on. Some of the storage plug-ins that are already supported in Citrix XenServer and Xen Cloud Platform (XCP) are:

1. NFS VHD: Storage repository (SR) plug-in that stores disks as Virtual Hard Disk (VHD) files on a remote Network File System (NFS).
2. Local VHD on LVM: SR plug-in that represents disks as VHD disks on Logical Volumes (LVM) within a locally-attached Volume Group.
3. HBA LUN-per-VDI driver: SR plug-in that represents Logical Units (LUs) as Virtual Disk Images (VDIs) sourced by host bus adapters (HBAs). For example, hardware-based iSCSI or FC support.
4. NetApp: SR driver for mapping of LUNs to VDIs on a NETAPP server, providing use of fast snapshot and clone features on the filer.
5. LVHD over FC: SR plug-in that represents disks as VHDs on Logical Volumes within a Volume Group created on an HBA LUN. For example, hardware-based iSCSI or FC support.
6. iSCSI: Base iSCSI SR driver, provides a LUN-per-VDI. Does not support creation of VDIs but accesses existing LUNs on a target.
7. LVHD over iSCSI: SR plug-in that represents disks as Logical Volumes within a Volume Group created on an iSCSI LUN.
8. EqualLogic: SR driver for mapping of LUNs to VDIs on a EQUALLOGIC array group, providing use of fast snapshot and clone features on the array.

Design and operation

Definitions

- **Back-end:** A term for a particular storage back-end. This could be iSCSI, NFS, NetApp, and so on.
- **Back-end-config:** All the parameters required to connect to a specific back-end. For example, for NFS, this would be the server, path, and so on.
- **Flavor:** This term is equivalent to volume "types". A user friendly term to specify some notion of quality of service. For example, "gold" might mean that the volumes use a back-end where backups are possible. A flavor can be associated with multiple back-ends. The volume scheduler, with the help of the driver, decides which back-end is used to create a volume of a particular flavor. Currently, the driver uses a simple "first-fit" policy, where the first back-end that can successfully create this volume is the one that is used.

Operation

The admin uses the `nova-manage` command detailed below to add flavors and back-ends.

One or more `cinder-volume` service instances are deployed for each availability zone. When an instance is started, it creates storage repositories (SRs) to connect to the back-ends available within that zone. All `cinder-volume` instances within a zone can see all the available back-ends. These instances are completely symmetric and hence should be able to service any `create_volume` request within the zone.



On XenServer, PV guests required

Note that when using XenServer you can only attach a volume to a PV guest.

Configure XenAPI Storage Manager

Prerequisites

1. `xensm` requires that you use either Citrix XenServer or XCP as the hypervisor. The NetApp and EqualLogic back-ends are not supported on XCP.
2. Ensure all `hosts` running volume and Compute services have connectivity to the storage system.

Configuration

- **Set the following configuration options for the nova volume service: (`nova-compute` also requires the `volume_driver` configuration option.)**

```
--volume_driver="nova.volume.xensm.XenSMDriver"  
--use_local_volumes=False
```

- **You must create the back-end configurations that the volume driver uses before you start the volume service.**

```
$ nova-manage sm flavor_create <label> <description>
$ nova-manage sm flavor_delete <label>
$ nova-manage sm backend_add <flavor label> <SR type> [config connection
parameters]
```



Note

SR type and configuration connection parameters are in keeping with the [XenAPI Command Line Interface](#).

```
$ nova-manage sm backend_delete <back-end-id>
```

Example: For the NFS storage manager plug-in, run these commands:

```
$ nova-manage sm flavor_create gold "Not all that glitters"
$ nova-manage sm flavor_delete gold
$ nova-manage sm backend_add gold nfs name_label=myback-end server=myserver
serverpath=/local/scratch/myname
$ nova-manage sm backend_remove 1
```

- Start `cinder-volume` and `nova-compute` with the new configuration options.

Create and access the volumes from VMs

Currently, the flavors have not been tied to the volume types API. As a result, we simply end up creating volumes in a "first fit" order on the given back-ends.

Use the standard `euca-*` or OpenStack API commands (such as volume extensions) to create, destroy, attach, or detach volumes.

Zadara

There is a volume back-end for Zadara. Set the following in your `cinder.conf`, and use the following options to configure it.

```
volume_driver=cinder.volume.drivers.zadara.ZadaraVPSAISCSDriver
```

Table 1.29. Description of configuration options for zadara

Configuration option = Default value	Description
[DEFAULT]	
zadara_password = None	(StrOpt) Password for the VPSA
zadara_user = None	(StrOpt) User name for the VPSA
zadara_vol_encrypt = False	(BoolOpt) Default encryption policy for volumes
zadara_vol_name_template = OS_%s	(StrOpt) Default template for VPSA volume names
zadara_vol_thin = True	(BoolOpt) Default thin provisioning policy for volumes
zadara_vpsa_allow_nonexistent_delete = True	(BoolOpt) Don't halt on deletion of non-existing volumes
zadara_vpsa_auto_detach_on_delete = True	(BoolOpt) Automatically detach from servers on volume delete
zadara_vpsa_ip = None	(StrOpt) Management IP of Zadara VPSA
zadara_vpsa_poolname = None	(StrOpt) Name of VPSA storage pool for volumes

Configuration option = Default value	Description
zadara_vpsa_port = None	(StrOpt) Zadara VPSA port number
zadara_vpsa_use_ssl = False	(BoolOpt) Use SSL connection

Sheepdog driver

Sheepdog is an open-source distributed storage system that provides a virtual storage pool utilizing internal disk of commodity servers.

Sheepdog scales to several hundred nodes, and has powerful virtual disk management features like snapshot, cloning, rollback, thin provisioning.

More information can be found on [Sheepdog Project](#).

This driver enables use of Sheepdog through Qemu/KVM.

Set the following `volume_driver` in `cinder.conf`:

```
volume_driver=cinder.volume.drivers.sheepdog.SheepdogDriver
```

Backup drivers

This section describes how to configure the `cinder-backup` service and its drivers.

The volume drivers are included with the Block Storage repository (<https://github.com/openstack/cinder>). To set a backup driver, use the `backup_driver` flag. By default there is no backup driver enabled.

Ceph backup driver

The Ceph backup driver backs up volumes of any type to a Ceph back-end store. The driver can also detect whether the volume to be backed up is a Ceph RBD volume, and if so, it tries to perform incremental and differential backups.

For source Ceph RBD volumes, you can perform backups within the same Ceph pool (not recommended) and backups between different Ceph pools and between different Ceph clusters.

At the time of writing, differential backup support in Ceph/librbd was quite new. This driver attempts a differential backup in the first instance. If the differential backup fails, the driver falls back to full backup/copy.

If incremental backups are used, multiple backups of the same volume are stored as snapshots so that minimal space is consumed in the backup store. It takes far less time to restore a volume than to take a full copy.



Note

Block Storage enables you to:

- Restore to a new volume, which is the default and recommended action.

- Restore to the original volume from which the backup was taken. The restore action takes a full copy because this is the safest action.

To enable the Ceph backup driver, include the following option in the `cinder.conf` file:

```
backup_driver = cinder.backup.drivers.ceph
```

The following configuration options are available for the Ceph backup driver.

Table 1.30. Description of configuration options for backups_ceph

Configuration option = Default value	Description
[DEFAULT]	
backup_ceph_chunk_size = 134217728	(IntOpt) The chunk size, in bytes, that a backup is broken into before transfer to the Ceph object store.
backup_ceph_conf = /etc/ceph/ceph.conf	(StrOpt) Ceph configuration file to use.
backup_ceph_pool = backups	(StrOpt) The Ceph pool where volume backups are stored.
backup_ceph_stripe_count = 0	(IntOpt) RBD stripe count to use when creating a backup image.
backup_ceph_stripe_unit = 0	(IntOpt) RBD stripe unit to use when creating a backup image.
backup_ceph_user = cinder	(StrOpt) The Ceph user to connect with. Default here is to use the same user as for Cinder volumes. If not using cephx this should be set to None.
restore_discard_excess_bytes = True	(BoolOpt) If True, always discard excess bytes when restoring volumes i.e. pad with zeroes.

This example shows the default options for the Ceph backup driver.

```
backup_ceph_conf=/etc/ceph/ceph.conf
backup_ceph_user = cinder
backup_ceph_chunk_size = 134217728
backup_ceph_pool = backups
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
```

IBM Tivoli Storage Manager backup driver

The IBM Tivoli Storage Manager (TSM) backup driver enables performing volume backups to a TSM server.

The TSM client should be installed and configured on the machine running the `cinder-backup` service. See the *IBM Tivoli Storage Manager Backup-Archive Client Installation and User's Guide* for details on installing the TSM client.

To enable the IBM TSM backup driver, include the following option in `cinder.conf`:

```
backup_driver = cinder.backup.drivers.tsm
```

The following configuration options are available for the TSM backup driver.

Table 1.31. Description of configuration options for backups_tsm

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
backup_tsm_compression = True	(BoolOpt) Enable or Disable compression for backups
backup_tsm_password = password	(StrOpt) TSM password for the running username
backup_tsm_volume_prefix = backup	(StrOpt) Volume prefix for the backup id when backing up to TSM

This example shows the default options for the TSM backup driver.

```
backup_tsm_volume_prefix = backup
backup_tsm_password = password
backup_tsm_compression = True
```

Swift backup driver

The backup driver for Swift back-end performs a volume backup to a Swift object storage system.

To enable the Swift backup driver, include the following option in the `cinder.conf` file:

```
backup_driver = cinder.backup.drivers.swift
```

The following configuration options are available for the Swift back-end backup driver.

Table 1.32. Description of configuration options for backups_swift

Configuration option = Default value	Description
[DEFAULT]	
backup_swift_auth = per_user	(StrOpt) Swift authentication mechanism
backup_swift_container = volumebackups	(StrOpt) The default Swift container to use
backup_swift_key = None	(StrOpt) Swift key for authentication
backup_swift_object_size = 52428800	(IntOpt) The size in bytes of Swift backup objects
backup_swift_retry_attempts = 3	(IntOpt) The number of retries to make for Swift operations
backup_swift_retry_backoff = 2	(IntOpt) The backoff time in seconds between Swift retries
backup_swift_url = http://localhost:8080/v1/AUTH_	(StrOpt) The URL of the Swift endpoint
backup_swift_user = None	(StrOpt) Swift user name

This example shows the default options for the Swift back-end backup driver.

```
backup_swift_url = http://localhost:8080/v1/AUTH
backup_swift_auth = per_user
backup_swift_user = <None>
backup_swift_key = <None>
backup_swift_container = volumebackups
backup_swift_object_size = 52428800
backup_swift_retry_attempts = 3
backup_swift_retry_backoff = 2
backup_compression_algorithm = zlib
```

Block Storage sample configuration files

All the files in this section can be found in `/etc/cinder`.

cinder.conf

Use the `cinder.conf` file to configure the majority of the Block Storage service options.

```
[DEFAULT]

#
# Options defined in oslo.messaging
#

# Use durable queues in AMQP. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_durable_queues
#amqp_durable_queues=false

# Auto-delete queues in AMQP. (boolean value)
#amqp_auto_delete=false

# Size of RPC connection pool. (integer value)
#rpc_conn_pool_size=30

# Qpid broker hostname. (string value)
#qpid_hostname=localhost

# Qpid broker port. (integer value)
#qpid_port=5672

# Qpid HA cluster host:port pairs. (list value)
#qpid_hosts=$qpid_hostname:$qpid_port

# Username for Qpid connection. (string value)
#qpid_username=

# Password for Qpid connection. (string value)
#qpid_password=

# Space separated list of SASL mechanisms to use for auth.
# (string value)
#qpid_sasl_mechanisms=

# Seconds between connection keepalive heartbeats. (integer
# value)
#qpid_heartbeat=60

# Transport to use, either 'tcp' or 'ssl'. (string value)
#qpid_protocol=tcp

# Whether to disable the Nagle algorithm. (boolean value)
#qpid_tcp_nodelay=true

# The number of prefetched messages held by receiver. (integer
# value)
#qpid_receiver_capacity=1

# The qpid topology version to use. Version 1 is what was
# originally used by impl_qpid. Version 2 includes some
# backwards-incompatible changes that allow broker federation
# to work. Users should update to version 2 when they are
# able to take everything down, as it requires a clean break.
```

```
# (integer value)
#qpidd_topology_version=1

# SSL version to use (valid only if SSL enabled). valid values
# are TLSv1 and SSLv23. SSLv2 and SSLv3 may be available on
# some distributions. (string value)
#kombu_ssl_version=

# SSL key file (valid only if SSL enabled). (string value)
#kombu_ssl_keyfile=

# SSL cert file (valid only if SSL enabled). (string value)
#kombu_ssl_certfile=

# SSL certification authority file (valid only if SSL
# enabled). (string value)
#kombu_ssl_ca_certs=

# How long to wait before reconnecting in response to an AMQP
# consumer cancel notification. (floating point value)
#kombu_reconnect_delay=1.0

# The RabbitMQ broker address where a single node is used.
# (string value)
#rabbit_host=localhost

# The RabbitMQ broker port where a single node is used.
# (integer value)
#rabbit_port=5672

# RabbitMQ HA cluster host:port pairs. (list value)
#rabbit_hosts=$rabbit_host:$rabbit_port

# Connect over SSL for RabbitMQ. (boolean value)
#rabbit_use_ssl=false

# The RabbitMQ userid. (string value)
#rabbit_userid=guest

# The RabbitMQ password. (string value)
#rabbit_password=guest

# The RabbitMQ login method. (string value)
#rabbit_login_method=AMQPPLAIN

# The RabbitMQ virtual host. (string value)
#rabbit_virtual_host=/

# How frequently to retry connecting with RabbitMQ. (integer
# value)
#rabbit_retry_interval=1

# How long to backoff for between retries when connecting to
# RabbitMQ. (integer value)
#rabbit_retry_backoff=2

# Maximum number of RabbitMQ connection retries. Default is 0
# (infinite retry count). (integer value)
#rabbit_max_retries=0
```

```
# Use HA queues in RabbitMQ (x-ha-policy: all). If you change
# this option, you must wipe the RabbitMQ database. (boolean
# value)
#rabbit_ha_queues=false

# Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
# (boolean value)
#fake_rabbit=false

# ZeroMQ bind address. Should be a wildcard (*), an ethernet
# interface, or IP. The "host" option should point or resolve
# to this address. (string value)
#rpc_zmq_bind_address=*

# MatchMaker driver. (string value)
#rpc_zmq_matchmaker=oslo.messaging._drivers.matchmaker.MatchMakerLocalhost

# ZeroMQ receiver listening port. (integer value)
#rpc_zmq_port=9501

# Number of ZeroMQ contexts, defaults to 1. (integer value)
#rpc_zmq_contexts=1

# Maximum number of ingress messages to locally buffer per
# topic. Default is unlimited. (integer value)
#rpc_zmq_topic_backlog=<None>

# Directory for holding IPC sockets. (string value)
#rpc_zmq_ipc_dir=/var/run/openstack

# Name of this node. Must be a valid hostname, FQDN, or IP
# address. Must match "host" option, if running Nova. (string
# value)
#rpc_zmq_host=cinder

# Seconds to wait before a cast expires (TTL). Only supported
# by impl_zmq. (integer value)
#rpc_cast_timeout=30

# Heartbeat frequency. (integer value)
#matchmaker_heartbeat_freq=300

# Heartbeat time-to-live. (integer value)
#matchmaker_heartbeat_ttl=600

# Size of RPC greenthread pool. (integer value)
#rpc_thread_pool_size=64

# Driver or drivers to handle sending notifications. (multi
# valued)
#notification_driver=

# AMQP topic used for OpenStack notifications. (list value)
# Deprecated group/name - [rpc_notifier2]/topics
#notification_topics=notifications

# Seconds to wait for a response from a call. (integer value)
#rpc_response_timeout=60

# A URL representing the messaging driver to use and its full
```

```
# configuration. If not set, we fall back to the rpc_backend
# option and driver specific configuration. (string value)
#transport_url=<None>

# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
#rpc_backend=rabbit

# The default exchange under which topics are scoped. May be
# overridden by an exchange name specified in the
# transport_url option. (string value)
#control_exchange=openstack

#
# Options defined in cinder.exception
#
# make exception message format errors fatal (boolean value)
#fatal_exception_format_errors=false

#
# Options defined in cinder.policy
#
# JSON file representing policy (string value)
#policy_file=policy.json

# Rule checked when requested rule is not found (string value)
#policy_default_rule=default

#
# Options defined in cinder.quota
#
# number of volumes allowed per project (integer value)
#quota_volumes=10

# number of volume snapshots allowed per project (integer
# value)
#quota_snapshots=10

# number of volume gigabytes (snapshots are also included)
# allowed per project (integer value)
#quota_gigabytes=1000

# number of seconds until a reservation expires (integer
# value)
#reservation_expire=86400

# count of reservations until usage is refreshed (integer
# value)
#until_refresh=0

# number of seconds between subsequent usage refreshes
# (integer value)
#max_age=0
```

```
# default driver to use for quota checks (string value)
#quota_driver=cinder.quota.DbQuotaDriver

# whether to use default quota class for default quota
# (boolean value)
#use_default_quota_class=true

#
# Options defined in cinder.service
#

# seconds between nodes reporting state to datastore (integer
# value)
#report_interval=10

# seconds between running periodic tasks (integer value)
#periodic_interval=60

# range of seconds to randomly delay when starting the
# periodic task scheduler to reduce stampeding. (Disable by
# setting to 0) (integer value)
#periodic_fuzzy_delay=60

# IP address for OpenStack Volume API to listen (string value)
#osapi_volume_listen=0.0.0.0

# port for os volume api to listen (integer value)
#osapi_volume_listen_port=8776

# Number of workers for OpenStack Volume API service (integer
# value)
#osapi_volume_workers=<None>

#
# Options defined in cinder.test
#

# File name of clean sqlite db (string value)
#sqlite_clean_db=clean.sqlite

#
# Options defined in cinder.wsgi
#

# Maximum line size of message headers to be accepted.
# max_header_line may need to be increased when using large
# tokens (typically those generated by the Keystone v3 API
# with big service catalogs). (integer value)
#max_header_line=16384

# If False, closes the client socket connection explicitly.
# Setting it to True to maintain backward compatibility.
# Recommended setting is set it to False. (boolean value)
#wsgi_keep_alive=true

# Sets the value of TCP_KEEPIIDLE in seconds for each server
# socket. Not supported on OS X. (integer value)
```

```
#tcp_keepidle=600

# CA certificate file to use to verify connecting clients
# (string value)
#ssl_ca_file=<None>

# Certificate file to use when starting the server securely
# (string value)
#ssl_cert_file=<None>

# Private key file to use when starting the server securely
# (string value)
#ssl_key_file=<None>

#
# Options defined in cinder.api.common
#

# the maximum number of items returned in a single response
# from a collection resource (integer value)
#osapi_max_limit=1000

# Base URL that will be presented to users in links to the
# OpenStack Volume API (string value)
# Deprecated group/name - [DEFAULT]/osapi_compute_link_prefix
#osapi_volume_base_URL=<None>

#
# Options defined in cinder.api.middleware.auth
#

# Treat X-Forwarded-For as the canonical remote address. Only
# enable this if you have a sanitizing proxy. (boolean value)
#use_forwarded_for=false

#
# Options defined in cinder.api.middleware.sizelimit
#

# Max size for body of a request (integer value)
#osapi_max_request_body_size=114688

#
# Options defined in cinder.backup.driver
#

# Backup metadata version to be used when backing up volume
# metadata. If this number is bumped, make sure the service
# doing the restore supports the new version. (integer value)
#backup_metadata_version=1

#
# Options defined in cinder.backup.drivers.ceph
#
```



```
# Ceph configuration file to use. (string value)
#backup_ceph_conf=/etc/ceph/ceph.conf

# The Ceph user to connect with. Default here is to use the
# same user as for Cinder volumes. If not using cephx this
# should be set to None. (string value)
#backup_ceph_user=cinder

# The chunk size, in bytes, that a backup is broken into
# before transfer to the Ceph object store. (integer value)
#backup_ceph_chunk_size=134217728

# The Ceph pool where volume backups are stored. (string
# value)
#backup_ceph_pool=backups

# RBD stripe unit to use when creating a backup image.
# (integer value)
#backup_ceph_stripe_unit=0

# RBD stripe count to use when creating a backup image.
# (integer value)
#backup_ceph_stripe_count=0

# If True, always discard excess bytes when restoring volumes
# i.e. pad with zeroes. (boolean value)
#restore_discard_excess_bytes=true

#
# Options defined in cinder.backup.drivers.swift
#

# The URL of the Swift endpoint (string value)
#backup_swift_url=http://localhost:8080/v1/AUTH_

# Swift authentication mechanism (string value)
#backup_swift_auth=per_user

# Swift user name (string value)
#backup_swift_user=<None>

# Swift key for authentication (string value)
#backup_swift_key=<None>

# The default Swift container to use (string value)
#backup_swift_container=volumebackups

# The size in bytes of Swift backup objects (integer value)
#backup_swift_object_size=52428800

# The number of retries to make for Swift operations (integer
# value)
#backup_swift_retry_attempts=3

# The backoff time in seconds between Swift retries (integer
# value)
#backup_swift_retry_backoff=2

# Compression algorithm (None to disable) (string value)
```

```
#backup_compression_algorithm=zlib

#
# Options defined in cinder.backup.drivers.tsm
#

# Volume prefix for the backup id when backing up to TSM
# (string value)
#backup_tsm_volume_prefix=backup

# TSM password for the running username (string value)
#backup_tsm_password=password

# Enable or Disable compression for backups (boolean value)
#backup_tsm_compression=true

#
# Options defined in cinder.backup.manager
#

# Driver to use for backups. (string value)
# Deprecated group/name - [DEFAULT]/backup_service
#backup_driver=cinder.backup.drivers.swift

#
# Options defined in cinder.common.config
#

# File name for the paste.deploy config for cinder-api (string
# value)
#api_paste_config=api-paste.ini

# Top-level directory for maintaining cinder's state (string
# value)
# Deprecated group/name - [DEFAULT]/pybasedir
#state_path=/var/lib/cinder

# ip address of this host (string value)
#my_ip=10.0.0.1

# default glance hostname or ip (string value)
#glance_host=$my_ip

# default glance port (integer value)
#glance_port=9292

# A list of the glance api servers available to cinder
# ([hostname|ip]:port) (list value)
#glance_api_servers=$glance_host:$glance_port

# Version of the glance api to use (integer value)
#glance_api_version=1

# Number retries when downloading an image from glance
# (integer value)
#glance_num_retries=0
```

```
# Allow to perform insecure SSL (https) requests to glance
# (boolean value)
#glance_api_insecure=false

# Whether to attempt to negotiate SSL layer compression when
# using SSL (https) requests. Set to False to disable SSL
# layer compression. In some cases disabling this may improve
# data throughput, eg when high network bandwidth is available
# and you are using already compressed image formats such as
# qcow2 . (boolean value)
#glance_api_ssl_compression=false

# http/https timeout value for glance operations. If no value
# (None) is supplied here, the glanceclient default value is
# used. (integer value)
#glance_request_timeout=<None>

# the topic scheduler nodes listen on (string value)
#scheduler_topic=cinder-scheduler

# the topic volume nodes listen on (string value)
#volume_topic=cinder-volume

# the topic volume backup nodes listen on (string value)
#backup_topic=cinder-backup

# Deploy v1 of the Cinder API. (boolean value)
#enable_v1_api=true

# Deploy v2 of the Cinder API. (boolean value)
#enable_v2_api=true

# whether to rate limit the api (boolean value)
#api_rate_limit=true

# Specify list of extensions to load when using
# osapi_volume_extension option with
# cinder.api.contrib.select_extensions (list value)
#osapi_volume_ext_list=

# osapi volume extension to load (multi valued)
#osapi_volume_extension=cinder.api.contrib.standard_extensions

# full class name for the Manager for volume (string value)
#volume_manager=cinder.volume.manager.VolumeManager

# full class name for the Manager for volume backup (string
# value)
#backup_manager=cinder.backup.manager.BackupManager

# full class name for the Manager for scheduler (string value)
#scheduler_manager=cinder.scheduler.manager.SchedulerManager

# Name of this node. This can be an opaque identifier. It is
# not necessarily a hostname, FQDN, or IP address. (string
# value)
#host=cinder

# availability zone of this node (string value)
#storage_availability_zone=nova
```

```
# default availability zone to use when creating a new volume.
# If this is not set then we use the value from the
# storage_availability_zone option as the default
# availability_zone for new volumes. (string value)
#default_availability_zone=<None>

# default volume type to use (string value)
#default_volume_type=<None>

# time period to generate volume usages for. Time period must
# be hour, day, month or year (string value)
#volume_usage_audit_period=month

# Path to the rootwrap configuration file to use for running
# commands as root (string value)
#rootwrap_config=/etc/cinder/rootwrap.conf

# Enable monkey patching (boolean value)
#monkey_patch=false

# List of modules/decorators to monkey patch (list value)
#monkey_patch_modules=

# maximum time since last check-in for up service (integer
# value)
#service_down_time=60

# The full class name of the volume API class to use (string
# value)
#volume_api_class=cinder.volume.api.API

# The full class name of the volume backup API class (string
# value)
#backup_api_class=cinder.backup.api.API

# The strategy to use for auth. Supports noauth, keystone, and
# deprecated. (string value)
#auth_strategy=noauth

# A list of backend names to use. These backend names should
# be backed by a unique [CONFIG] group with its options (list
# value)
#enabled_backends=<None>

# Whether snapshots count against GigaByte quota (boolean
# value)
#no_snapshot_gb_quota=false

# The full class name of the volume transfer API class (string
# value)
#transfer_api_class=cinder.transfer.api.API

#
# Options defined in cinder.compute
#

# The full class name of the compute API class to use (string
# value)
```



```
#
# Options defined in cinder.db.base
#
# driver to use for database access (string value)
#db_driver=cinder.db

#
# Options defined in cinder.image.glance
#
# A list of url schemes that can be downloaded directly via
# the direct_url. Currently supported schemes: [file]. (list
# value)
#allowed_direct_url_schemes=

#
# Options defined in cinder.image.image_utils
#
# Directory used for temporary storage during image conversion
# (string value)
#image_conversion_dir=$state_path/conversion

#
# Options defined in cinder.openstack.common.db.sqlalchemy.session
#
# the filename to use with sqlite (string value)
#sqlite_db=cinder.sqlite

# If true, use synchronous mode for sqlite (boolean value)
#sqlite_synchronous=true

#
# Options defined in cinder.openstack.common.eventlet_backdoor
#
# 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
# service's log file. (string value)
#backdoor_port=<None>

#
# Options defined in cinder.openstack.common.lockutils
#
# Whether to disable inter-process locks (boolean value)
#disable_process_locking=false
```

```
# Directory to use for lock files. Default to a temp directory
# (string value)
#lock_path=<None>

#
# Options defined in cinder.openstack.common.log
#

# Print debugging output (set logging level to DEBUG instead
# of default WARNING level). (boolean value)
#debug=false

# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
#verbose=false

# Log output to standard error (boolean value)
#use_stderr=true

# Format string to use for log messages with context (string
# value)
#logging_context_format_string=%(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 without context
# (string value)
#logging_default_format_string=%(asctime)s.%(msecs)03d %(process)d
  %(levelname)s %(name)s [-] %(instance)s%(message)s

# Data to append to log format when level is DEBUG (string
# value)
#logging_debug_format_suffix=%(funcName)s %(pathname)s:%(lineno)d

# Prefix each line of exception output with this format
# (string value)
#logging_exception_prefix=%(asctime)s.%(msecs)03d %(process)d TRACE %(name)s
  %(instance)s

# List of logger=LEVEL pairs (list value)
#default_log_levels=amqp=WARN,amqpplib=WARN,boto=WARN,qpid=WARN,sqlalchemy=
  WARN,suds=INFO,oslo.messaging=INFO,iso8601=WARN,requests.packages.urllib3.
  connectionpool=WARN

# Publish error events (boolean value)
#publish_errors=false

# Make deprecations fatal (boolean value)
#fatal_deprecations=false

# If an instance is passed with the log message, format it
# like this (string value)
#instance_format="[instance: %(uuid)s] "

# If an instance UUID is passed with the log message, format
# it like this (string value)
#instance_uuid_format="[instance: %(uuid)s] "
```

```
# The name of logging configuration file. It does not disable
# existing loggers, but just appends specified logging
# configuration to any other existing logging options. Please
# see the Python logging module documentation for details on
# logging configuration files. (string value)
# Deprecated group/name - [DEFAULT]/log_config
#log_config_append=<None>

# DEPRECATED. A logging.Formatter log message format string
# which may use any of the available logging.LogRecord
# attributes. This option is deprecated. Please use
# logging_context_format_string and
# logging_default_format_string instead. (string value)
#log_format=<None>

# Format string for %(asctime)s in log records. Default:
# %(default)s (string value)
#log_date_format=%Y-%m-%d %H:%M:%S

# (Optional) Name of log file to output to. If no default is
# set, logging will go to stdout. (string value)
# Deprecated group/name - [DEFAULT]/logfile
#log_file=<None>

# (Optional) The base directory used for relative --log-file
# paths (string value)
# Deprecated group/name - [DEFAULT]/logdir
#log_dir=<None>

# Use syslog for logging. Existing syslog format is DEPRECATED
# during I, and then will be changed in J to honor RFC5424
# (boolean value)
#use_syslog=false

# (Optional) Use syslog rfc5424 format for logging. If
# enabled, will add APP-NAME (RFC5424) before the MSG part of
# the syslog message. The old format without APP-NAME is
# deprecated in I, and will be removed in J. (boolean value)
#use_syslog_rfc_format=false

# Syslog facility to receive log lines (string value)
#syslog_log_facility=LOG_USER

#
# Options defined in cinder.openstack.common.periodic_task
#

# Some periodic tasks can be run in a separate process. Should
# we run them here? (boolean value)
#run_external_periodic_tasks=true

#
# Options defined in cinder.scheduler.driver
#

# The scheduler host manager class to use (string value)
#scheduler_host_manager=cinder.scheduler.host_manager.HostManager
```



```
# Maximum number of attempts to schedule an volume (integer
# value)
#scheduler_max_attempts=3

#
# Options defined in cinder.scheduler.host_manager
#

# Which filter class names to use for filtering hosts when not
# specified in the request. (list value)
#scheduler_default_filters=AvailabilityZoneFilter,CapacityFilter,
CapabilitiesFilter

# Which weigher class names to use for weighing hosts. (list
# value)
#scheduler_default_weighers=CapacityWeigher

#
# Options defined in cinder.scheduler.manager
#

# Default scheduler driver to use (string value)
#scheduler_driver=cinder.scheduler.filter_scheduler.FilterScheduler

#
# Options defined in cinder.scheduler.scheduler_options
#

# Absolute path to scheduler configuration JSON file. (string
# value)
#scheduler_json_config_location=

#
# Options defined in cinder.scheduler.simple
#

# This configure option has been deprecated along with the
# SimpleScheduler. New scheduler is able to gather capacity
# information for each host, thus setting the maximum number
# of volume gigabytes for host is no longer needed. It's safe
# to remove this configure from cinder.conf. (integer value)
#max_gigabytes=10000

#
# Options defined in cinder.scheduler.weights.capacity
#

# Multiplier used for weighing volume capacity. Negative
# numbers mean to stack vs spread. (floating point value)
#capacity_weight_multiplier=1.0

# Multiplier used for weighing volume capacity. Negative
# numbers mean to stack vs spread. (floating point value)
#allocated_capacity_weight_multiplier=-1.0
```

```
#
# Options defined in cinder.transfer.api
#
# The number of characters in the salt. (integer value)
#volume_transfer_salt_length=8
# The number of characters in the autogenerated auth key.
# (integer value)
#volume_transfer_key_length=16
#
# Options defined in cinder.volume.api
#
# Create volume from snapshot at the host where snapshot
# resides (boolean value)
#snapshot_same_host=true
# Ensure that the new volumes are the same AZ as snapshot or
# source volume (boolean value)
#cloned_volume_same_az=true
#
# Options defined in cinder.volume.driver
#
# The maximum number of times to rescan iSER targetto find
# volume (integer value)
#num_iser_scan_tries=3
# The maximum number of iser target ids per host (integer
# value)
#iser_num_targets=100
# prefix for iser volumes (string value)
#iser_target_prefix=iqn.2010-10.org.openstack:
# The IP address that the iSER daemon is listening on (string
# value)
#iser_ip_address=$my_ip
# The port that the iSER daemon is listening on (integer
# value)
#iser_port=3260
# iser target user-land tool to use (string value)
#iser_helper=tgtadm
# number of times to attempt to run flakey shell commands
# (integer value)
#num_shell_tries=3
# The percentage of backend capacity is reserved (integer
# value)
#reserved_percentage=0
```

```
# The maximum number of iscsi target ids per host (integer
# value)
#iscsi_num_targets=100

# prefix for iscsi volumes (string value)
#iscsi_target_prefix=iqn.2010-10.org.openstack:

# The IP address that the iSCSI daemon is listening on (string
# value)
#iscsi_ip_address=$my_ip

# The port that the iSCSI daemon is listening on (integer
# value)
#iscsi_port=3260

# The maximum number of times to rescan targets to find volume
# (integer value)
# Deprecated group/name - [DEFAULT]/num_iscsi_scan_tries
#num_volume_device_scan_tries=3

# The backend name for a given driver implementation (string
# value)
#volume_backend_name=<None>

# Do we attach/detach volumes in cinder using multipath for
# volume to image and image to volume transfers? (boolean
# value)
#use_multipath_for_image_xfer=false

# Method used to wipe old voumes (valid options are: none,
# zero, shred) (string value)
#volume_clear=zero

# Size in MiB to wipe at start of old volumes. 0 => all
# (integer value)
#volume_clear_size=0

# The flag to pass to ionice to alter the i/o priority of the
# process used to zero a volume after deletion, for example
# "-c3" for idle only priority. (string value)
#volume_clear_ionice=<None>

# iscsi target user-land tool to use (string value)
#iscsi_helper=tgtadm

# Volume configuration file storage directory (string value)
#volumes_dir=$state_path/volumes

# IET configuration file (string value)
#iet_conf=/etc/iet/ietd.conf

# Comma-separated list of initiator IQNs allowed to connect to
# the iSCSI target. (From Nova compute nodes.) (string value)
#lio_initiator_iqns=

# Sets the behavior of the iSCSI target to either perform
# blockio or fileio optionally, auto can be set and Cinder
# will autodetect type of backing device (string value)
#iscsi_iotype=fileio
```

```
# The default block size used when copying/clearing volumes
# (string value)
#volume_dd_blocksize=1M

#
# Options defined in cinder.volume.drivers.block_device
#

# List of all available devices (list value)
#available_devices=

#
# Options defined in cinder.volume.drivers.coraid
#

# IP address of Coraid ESM (string value)
#coraid_esm_address=

# User name to connect to Coraid ESM (string value)
#coraid_user=admin

# Name of group on Coraid ESM to which coraid_user belongs
# (must have admin privilege) (string value)
#coraid_group=admin

# Password to connect to Coraid ESM (string value)
#coraid_password=password

# Volume Type key name to store ESM Repository Name (string
# value)
#coraid_repository_key=coraid_repository

#
# Options defined in cinder.volume.drivers.emc.emc_smis_common
#

# use this file for cinder emc plugin config data (string
# value)
#cinder_emc_config_file=/etc/cinder/cinder_emc_config.xml

#
# Options defined in cinder.volume.drivers.emc.emc_vnx_cli
#

# Naviseccli Path (string value)
#naviseccli_path=

# ISCSI pool name (string value)
#storage_vnx_pool_name=<None>

# Default Time Out For CLI operations in minutes (integer
# value)
#default_timeout=20

# Default max number of LUNs in a storage group (integer
# value)
```



```
#
# Options defined in cinder.volume.drivers.huawei
#
# config data for cinder huawei plugin (string value)
#cinder_huawei_conf_file=/etc/cinder/cinder_huawei_conf.xml
#
# Options defined in cinder.volume.drivers.ibm.gpfs
#
# Specifies the path of the GPFS directory where Block Storage
# volume and snapshot files are stored. (string value)
#gpfs_mount_point_base=<None>
#
# Specifies the path of the Image service repository in GPFS.
# Leave undefined if not storing images in GPFS. (string
# value)
#gpfs_images_dir=<None>
#
# Specifies the type of image copy to be used. Set this when
# the Image service repository also uses GPFS so that image
# files can be transferred efficiently from the Image service
# to the Block Storage service. There are two valid values:
# "copy" specifies that a full copy of the image is made;
# "copy_on_write" specifies that copy-on-write optimization
# strategy is used and unmodified blocks of the image file are
# shared efficiently. (string value)
#gpfs_images_share_mode=<None>
#
# Specifies an upper limit on the number of indirections
# required to reach a specific block due to snapshots or
# clones. A lengthy chain of copy-on-write snapshots or
# clones can have a negative impact on performance, but
# improves space utilization. 0 indicates unlimited clone
# depth. (integer value)
#gpfs_max_clone_depth=0
#
# Specifies that volumes are created as sparse files which
# initially consume no space. If set to False, the volume is
# created as a fully allocated file, in which case, creation
# may take a significantly longer time. (boolean value)
#gpfs_sparse_volumes=true
#
# Specifies the storage pool that volumes are assigned to. By
# default, the system storage pool is used. (string value)
#gpfs_storage_pool=<None>
#
# Options defined in cinder.volume.drivers.ibm.storwize_svc
#
# Storage system storage pool for volumes (string value)
#storwize_svc_volpool_name=volpool
#
# Storage system space-efficiency parameter for volumes
# (percentage) (integer value)
```

```
#storwize_svc_vol_rsize=2

# Storage system threshold for volume capacity warnings
# (percentage) (integer value)
#storwize_svc_vol_warning=0

# Storage system autoexpand parameter for volumes (True/False)
# (boolean value)
#storwize_svc_vol_autoexpand=true

# Storage system grain size parameter for volumes
# (32/64/128/256) (integer value)
#storwize_svc_vol_grainsize=256

# Storage system compression option for volumes (boolean
# value)
#storwize_svc_vol_compression=false

# Enable Easy Tier for volumes (boolean value)
#storwize_svc_vol_easytier=true

# The I/O group in which to allocate volumes (integer value)
#storwize_svc_vol_iogrp=0

# Maximum number of seconds to wait for FlashCopy to be
# prepared. Maximum value is 600 seconds (10 minutes) (integer
# value)
#storwize_svc_flashcopy_timeout=120

# Connection protocol (iSCSI/FC) (string value)
#storwize_svc_connection_protocol=iSCSI

# Configure CHAP authentication for iSCSI connections
# (Default: Enabled) (boolean value)
#storwize_svc_iscsi_chap_enabled=true

# Connect with multipath (FC only; iSCSI multipath is
# controlled by Nova) (boolean value)
#storwize_svc_multipath_enabled=false

# Allows vdisk to multi host mapping (boolean value)
#storwize_svc_multihostmap_enabled=true

#
# Options defined in cinder.volume.drivers.ibm.xiv_ds8k
#
# Proxy driver that connects to the IBM Storage Array (string
# value)
#xiv_ds8k_proxy=xiv_ds8k_openstack.nova_proxy.XIVDS8KNovaProxy

# Connection type to the IBM Storage Array
# (fibre_channel|iscsi) (string value)
#xiv_ds8k_connection_type=iscsi

# CHAP authentication mode, effective only for iscsi
# (disabled|enabled) (string value)
#xiv_chap=disabled
```

```
#
# Options defined in cinder.volume.drivers.lvm
#
# Name for the VG that will contain exported volumes (string
# value)
#volume_group=cinder-volumes
#
# If set, create lvms with multiple mirrors. Note that this
# requires lvm_mirrors + 2 pvs with available space (integer
# value)
#lvm_mirrors=0
#
# Type of LVM volumes to deploy; (default or thin) (string
# value)
#lvm_type=default
#
# Options defined in cinder.volume.drivers.netapp.options
#
# The vFiler unit on which provisioning of block storage
# volumes will be done. This option is only used by the driver
# when connecting to an instance with a storage family of Data
# ONTAP operating in 7-Mode. Only use this option when
# utilizing the MultiStore feature on the NetApp storage
# system. (string value)
#netapp_vfiler=<None>
#
# Administrative user account name used to access the storage
# system or proxy server. (string value)
#netapp_login=<None>
#
# Password for the administrative user account specified in
# the netapp_login option. (string value)
#netapp_password=<None>
#
# This option specifies the virtual storage server (Vserver)
# name on the storage cluster on which provisioning of block
# storage volumes should occur. If using the NFS storage
# protocol, this parameter is mandatory for storage service
# catalog support (utilized by Cinder volume type extra_specs
# support). If this option is specified, the exports belonging
# to the Vserver will only be used for provisioning in the
# future. Block storage volumes on exports not belonging to
# the Vserver specified by this option will continue to
# function normally. (string value)
#netapp_vserver=<None>
#
# The hostname (or IP address) for the storage system or proxy
# server. (string value)
#netapp_server_hostname=<None>
#
# The TCP port to use for communication with the storage
# system or proxy server. Traditionally, port 80 is used for
# HTTP and port 443 is used for HTTPS; however, this value
# should be changed if an alternate port has been configured
# on the storage system or proxy server. (integer value)
```



```
#netapp_server_port=80

# This option is used to specify the path to the E-Series
# proxy application on a proxy server. The value is combined
# with the value of the netapp_transport_type,
# netapp_server_hostname, and netapp_server_port options to
# create the URL used by the driver to connect to the proxy
# application. (string value)
#netapp_webservice_path=/devmgr/v2

# This option is only utilized when the storage family is
# configured to eseries. This option is used to restrict
# provisioning to the specified controllers. Specify the value
# of this option to be a comma separated list of controller
# hostnames or IP addresses to be used for provisioning.
# (string value)
#netapp_controller_ips=<None>

# Password for the NetApp E-Series storage array. (string
# value)
#netapp_sa_password=<None>

# This option is used to restrict provisioning to the
# specified storage pools. Only dynamic disk pools are
# currently supported. Specify the value of this option to be
# a comma separated list of disk pool names to be used for
# provisioning. (string value)
#netapp_storage_pools=<None>

# This option is used to define how the controllers in the
# E-Series storage array will work with the particular
# operating system on the hosts that are connected to it.
# (string value)
#netapp_eseries_host_type=linux_dm_mp

# If the percentage of available space for an NFS share has
# dropped below the value specified by this option, the NFS
# image cache will be cleaned. (integer value)
#thres_avl_size_perc_start=20

# When the percentage of available space on an NFS share has
# reached the percentage specified by this option, the driver
# will stop clearing files from the NFS image cache that have
# not been accessed in the last M minutes, where M is the
# value of the expiry_thres_minutes configuration option.
# (integer value)
#thres_avl_size_perc_stop=60

# This option specifies the threshold for last access time for
# images in the NFS image cache. When a cache cleaning cycle
# begins, images in the cache that have not been accessed in
# the last M minutes, where M is the value of this parameter,
# will be deleted from the cache to create free space on the
# NFS share. (integer value)
#expiry_thres_minutes=720

# This option specifies the path of the NetApp copy offload
# tool binary. Ensure that the binary has execute permissions
# set which allow the effective user of the cinder-volume
# process to execute the file. (string value)
```

```
#netapp_copyoffload_tool_path=<None>

# The quantity to be multiplied by the requested volume size
# to ensure enough space is available on the virtual storage
# server (Vserver) to fulfill the volume creation request.
# (floating point value)
#netapp_size_multiplier=1.2

# This option is only utilized when the storage protocol is
# configured to use iSCSI. This option is used to restrict
# provisioning to the specified controller volumes. Specify
# the value of this option to be a comma separated list of
# NetApp controller volume names to be used for provisioning.
# (string value)
#netapp_volume_list=<None>

# The storage family type used on the storage system; valid
# values are ontap_7mode for using Data ONTAP operating in
# 7-Mode, ontap_cluster for using clustered Data ONTAP, or
# eseries for using E-Series. (string value)
#netapp_storage_family=ontap_cluster

# The storage protocol to be used on the data path with the
# storage system; valid values are iscsi or nfs. (string
# value)
#netapp_storage_protocol=<None>

# The transport protocol used when communicating with the
# storage system or proxy server. Valid values are http or
# https. (string value)
#netapp_transport_type=http

#
# Options defined in cinder.volume.drivers.nexenta.options
#

# IP address of Nexenta SA (string value)
#nexenta_host=

# HTTP port to connect to Nexenta REST API server (integer
# value)
#nexenta_rest_port=2000

# Use http or https for REST connection (default auto) (string
# value)
#nexenta_rest_protocol=auto

# User name to connect to Nexenta SA (string value)
#nexenta_user=admin

# Password to connect to Nexenta SA (string value)
#nexenta_password=nexenta

# Nexenta target portal port (integer value)
#nexenta_iscsi_target_portal_port=3260

# pool on SA that will hold all volumes (string value)
#nexenta_volume=cinder
```

```
# IQN prefix for iSCSI targets (string value)
#nexenta_target_prefix=iqn.1986-03.com.sun:02:cinder-

# prefix for iSCSI target groups on SA (string value)
#nexenta_target_group_prefix=cinder/

# File with the list of available nfs shares (string value)
#nexenta_shares_config=/etc/cinder/nfs_shares

# Base dir containing mount points for nfs shares (string
# value)
#nexenta_mount_point_base=$state_path/mnt

# Create volumes as sparsed files which take no space.If set
# to False volume is created as regular file.In such case
# volume creation takes a lot of time. (boolean value)
#nexenta_sparsed_volumes=true

# Default compression value for new ZFS folders. (string
# value)
#nexenta_volume_compression=on

# If set True cache NexentaStor appliance volroot option
# value. (boolean value)
#nexenta_nms_cache_volroot=true

# Enable stream compression, level 1..9. 1 - gives best speed;
# 9 - gives best compression. (integer value)
#nexenta_rrmgr_compression=0

# TCP Buffer size in KiloBytes. (integer value)
#nexenta_rrmgr_tcp_buf_size=4096

# Number of TCP connections. (integer value)
#nexenta_rrmgr_connections=2

# block size for volumes (blank=default,8KB) (string value)
#nexenta_blocksize=

# flag to create sparse volumes (boolean value)
#nexenta_sparse=false

#
# Options defined in cinder.volume.drivers.nfs
#

# IP address or Hostname of NAS system. (string value)
#nas_ip=

# User name to connect to NAS system. (string value)
#nas_login=admin

# Password to connect to NAS system. (string value)
#nas_password=

# SSH port to use to connect to NAS system. (integer value)
#nas_ssh_port=22

# Filename of private key to use for SSH authentication.
```

```
# (string value)
#nas_private_key=

# File with the list of available nfs shares (string value)
#nfs_shares_config=/etc/cinder/nfs_shares

# Create volumes as sparsed files which take no space.If set
# to False volume is created as regular file.In such case
# volume creation takes a lot of time. (boolean value)
#nfs_sparsed_volumes=true

# Percent of ACTUAL usage of the underlying volume before no
# new volumes can be allocated to the volume destination.
# (floating point value)
#nfs_used_ratio=0.95

# This will compare the allocated to available space on the
# volume destination. If the ratio exceeds this number, the
# destination will no longer be valid. (floating point value)
#nfs_oversub_ratio=1.0

# Base dir containing mount points for nfs shares. (string
# value)
#nfs_mount_point_base=$state_path/mnt

# Mount options passed to the nfs client. See section of the
# nfs man page for details. (string value)
#nfs_mount_options=<None>

#
# Options defined in cinder.volume.drivers.rbd
#
# the RADOS pool in which rbd volumes are stored (string
# value)
#rbd_pool=rbd

# the RADOS client name for accessing rbd volumes - only set
# when using cephx authentication (string value)
#rbd_user=<None>

# path to the ceph configuration file to use (string value)
#rbd_ceph_conf=

# flatten volumes created from snapshots to remove dependency
# (boolean value)
#rbd_flatten_volume_from_snapshot=false

# the libvirt uuid of the secret for the rbd_uservolumes
# (string value)
#rbd_secret_uuid=<None>

# where to store temporary image files if the volume driver
# does not write them directly to the volume (string value)
#volume_tmp_dir=<None>

# maximum number of nested clones that can be taken of a
# volume before enforcing a flatten prior to next clone. A
# value of zero disables cloning (integer value)
```

```
#rbd_max_clone_depth=5

# Timeout value (in seconds) used when connecting to ceph
# cluster. If value < 0, no timeout is set and default
# librados value is used. (integer value)
#rados_connect_timeout=-1

#
# Options defined in cinder.volume.drivers.san.hp.hp_3par_common
#

# 3PAR WSAPI Server Url like https://<3par ip>:8080/api/v1
# (string value)
#hp3par_api_url=

# 3PAR Super user username (string value)
#hp3par_username=

# 3PAR Super user password (string value)
#hp3par_password=

# The CPG to use for volume creation (string value)
#hp3par_cpg=OpenStack

# The CPG to use for Snapshots for volumes. If empty
# hp3par_cpg will be used (string value)
#hp3par_cpg_snap=

# The time in hours to retain a snapshot. You can't delete it
# before this expires. (string value)
#hp3par_snapshot_retention=

# The time in hours when a snapshot expires and is deleted.
# This must be larger than expiration (string value)
#hp3par_snapshot_expiration=

# Enable HTTP debugging to 3PAR (boolean value)
#hp3par_debug=false

# List of target iSCSI addresses to use. (list value)
#hp3par_iscsi_ips=

#
# Options defined in cinder.volume.drivers.san.hp.hp_leftHand_rest_proxy
#

# HP LeftHand WSAPI Server Url like https://<LeftHand
# ip>:8081/lhos (string value)
#hplefthand_api_url=<None>

# HP LeftHand Super user username (string value)
#hplefthand_username=<None>

# HP LeftHand Super user password (string value)
#hplefthand_password=<None>

# HP LeftHand cluster name (string value)
#hplefthand_clustername=<None>
```

```
# Configure CHAP authentication for iSCSI connections
# (Default: Disabled) (boolean value)
#hplefthand_iscsi_chap_enabled=false

# Enable HTTP debugging to LeftHand (boolean value)
#hplefthand_debug=false

#
# Options defined in cinder.volume.drivers.san.hp.hp_msa_common
#

# The VDisk to use for volume creation. (string value)
#msa_vdisk=OpenStack

#
# Options defined in cinder.volume.drivers.san.san
#

# Use thin provisioning for SAN volumes? (boolean value)
#san_thin_provision=true

# IP address of SAN controller (string value)
#san_ip=

# Username for SAN controller (string value)
#san_login=admin

# Password for SAN controller (string value)
#san_password=

# Filename of private key to use for SSH authentication
# (string value)
#san_private_key=

# Cluster name to use for creating volumes (string value)
#san_clustername=

# SSH port to use with SAN (integer value)
#san_ssh_port=22

# Execute commands locally instead of over SSH; use if the
# volume service is running on the SAN device (boolean value)
#san_is_local=false

# SSH connection timeout in seconds (integer value)
#ssh_conn_timeout=30

# Minimum ssh connections in the pool (integer value)
#ssh_min_pool_conn=1

# Maximum ssh connections in the pool (integer value)
#ssh_max_pool_conn=5

#
# Options defined in cinder.volume.drivers.san.solaris
#
```

```
# The ZFS path under which to create zvols for volumes.
# (string value)
#san_zfs_volume_base=rpool/

#
# Options defined in cinder.volume.drivers.scality
#

# Path or URL to Scality SOFS configuration file (string
# value)
#scality_sofs_config=<None>

# Base dir where Scality SOFS shall be mounted (string value)
#scality_sofs_mount_point=$state_path/scality

# Path from Scality SOFS root to volume dir (string value)
#scality_sofs_volume_dir=cinder/volumes

#
# Options defined in cinder.volume.drivers.solidfire
#

# Set 512 byte emulation on volume creation; (boolean value)
#sf_emulate_512=true

# Allow tenants to specify QOS on create (boolean value)
#sf_allow_tenant_qos=false

# Create SolidFire accounts with this prefix. Any string can
# be used here, but the string "hostname" is special and will
# create a prefix using the cinder node hostname (previous
# default behavior). The default is NO prefix. (string value)
#sf_account_prefix=<None>

# SolidFire API port. Useful if the device api is behind a
# proxy on a different port. (integer value)
#sf_api_port=443

#
# Options defined in cinder.volume.drivers.vmware.vmdk
#

# IP address for connecting to VMware ESX/VC server. (string
# value)
#vmware_host_ip=<None>

# Username for authenticating with VMware ESX/VC server.
# (string value)
#vmware_host_username=<None>

# Password for authenticating with VMware ESX/VC server.
# (string value)
#vmware_host_password=<None>

# Optional VIM service WSDL Location e.g
# http://<server>/vimService.wsdl. Optional over-ride to
```

```
# default location for bug work-arounds. (string value)
#vmware_wsdl_location=<None>

# Number of times VMware ESX/VC server API must be retried
# upon connection related issues. (integer value)
#vmware_api_retry_count=10

# The interval (in seconds) for polling remote tasks invoked
# on VMware ESX/VC server. (integer value)
#vmware_task_poll_interval=5

# Name for the folder in the VC datacenter that will contain
# cinder volumes. (string value)
#vmware_volume_folder=cinder-volumes

# Timeout in seconds for VMDK volume transfer between Cinder
# and Glance. (integer value)
#vmware_image_transfer_timeout_secs=7200

# Max number of objects to be retrieved per batch. Query
# results will be obtained in batches from the server and not
# in one shot. Server may still limit the count to something
# less than the configured value. (integer value)
#vmware_max_objects_retrieval=100

# Optional string specifying the VMware VC server version. The
# driver attempts to retrieve the version from VMware VC
# server. Set this configuration only if you want to override
# the VC server version. (string value)
#vmware_host_version=<None>

#
# Options defined in cinder.volume.drivers.windows.windows
#

# Path to store VHD backed volumes (string value)
#windows_iscsi_lun_path=C:\iSCSIVirtualDisks

#
# Options defined in cinder.volume.drivers.xenapi.sm
#

# NFS server to be used by XenAPI NFS Driver (string value)
#xenapi_nfs_server=<None>

# Path of exported NFS, used by XenAPI NFS Driver (string value)
#xenapi_nfs_serverpath=<None>

# URL for XenAPI connection (string value)
#xenapi_connection_url=<None>

# Username for XenAPI connection (string value)
#xenapi_connection_username=root

# Password for XenAPI connection (string value)
#xenapi_connection_password=<None>

# Base path to the storage repository (string value)
```



```
#xenapi_sr_base_path=/var/run/sr-mount

#
# Options defined in cinder.volume.drivers.zadara
#

# Management IP of Zadara VPSA (string value)
#zadara_vpsa_ip=<None>

# Zadara VPSA port number (string value)
#zadara_vpsa_port=<None>

# Use SSL connection (boolean value)
#zadara_vpsa_use_ssl=false

# User name for the VPSA (string value)
#zadara_user=<None>

# Password for the VPSA (string value)
#zadara_password=<None>

# Name of VPSA storage pool for volumes (string value)
#zadara_vpsa_poolname=<None>

# Default thin provisioning policy for volumes (boolean value)
#zadara_vol_thin=true

# Default encryption policy for volumes (boolean value)
#zadara_vol_encrypt=false

# Default template for VPSA volume names (string value)
#zadara_vol_name_template=OS_%s

# Automatically detach from servers on volume delete (boolean
# value)
#zadara_vpsa_auto_detach_on_delete=true

# Don't halt on deletion of non-existing volumes (boolean
# value)
#zadara_vpsa_allow_nonexistent_delete=true

#
# Options defined in cinder.volume.manager
#

# Driver to use for volume creation (string value)
#volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver

# Timeout for creating the volume to migrate to when
# performing volume migration (seconds) (integer value)
#migration_create_volume_timeout_secs=300

# Offload pending volume delete during volume service startup
# (boolean value)
#volume_service_inithost_offload=false

# FC Zoning mode configured (string value)
#zoning_mode=none
```

```
# User defined capabilities, a JSON formatted string
# specifying key/value pairs. (string value)
#extra_capabilities={}

[BRCD_FABRIC_EXAMPLE]

#
# Options defined in cinder.zonemanager.drivers.brocade.brcd_fabric_opts
#

# Management IP of fabric (string value)
#fc_fabric_address=

# Fabric user ID (string value)
#fc_fabric_user=

# Password for user (string value)
#fc_fabric_password=

# Connecting port (integer value)
#fc_fabric_port=22

# overridden zoning policy (string value)
#zoning_policy=initiator-target

# overridden zoning activation state (boolean value)
#zone_activate=true

# overridden zone name prefix (string value)
#zone_name_prefix=<None>

# Principal switch WWN of the fabric (string value)
#principal_switch_wwn=<None>

[database]

#
# Options defined in cinder.openstack.common.db.api
#

# The backend to use for db (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend=sqlalchemy

# Enable the experimental use of thread pooling for all DB API
# calls (boolean value)
# Deprecated group/name - [DEFAULT]/dbapi_use_tpool
#use_tpool=false

#
# Options defined in cinder.openstack.common.db.sqlalchemy.session
#

# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
```

```
#connection=sqlite:///state_path/sqlite_db

# timeout before idle sql connections are reaped (integer
# value)
# Deprecated group/name - [DEFAULT]/sql_idle_timeout
#idle_timeout=3600

# Minimum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_min_pool_size
#min_pool_size=1

# Maximum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_pool_size
#max_pool_size=5

# maximum db connection retries during startup. (setting -1
# implies an infinite retry count) (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_retries
#max_retries=10

# interval between retries of opening a sql connection
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_retry_interval
#retry_interval=10

# If set, use this value for max_overflow with sqlalchemy
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_overflow
#max_overflow=<None>

# Verbosity of SQL debugging information. 0=None,
# 100=Everything (integer value)
# Deprecated group/name - [DEFAULT]/sql_connection_debug
#connection_debug=0

# Add python stack traces to SQL as comment strings (boolean
# value)
# Deprecated group/name - [DEFAULT]/sql_connection_trace
#connection_trace=false

[fc-zone-manager]

#
# Options defined in cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver
#

# Southbound connector for zoning operation (string value)
#brcd_sb_connector=cinder.zonemanager.drivers.brocade.brcd_fc_zone_client_cli.
BrcdFCZoneClientCLI

#
# Options defined in cinder.zonemanager.fc_zone_manager
#

# FC Zone Driver responsible for zone management (string
# value)
```

```
#zone_driver=cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver.
BrcdFCZoneDriver

# Zoning policy configured by user (string value)
#zoning_policy=initiator-target

# Comma separated list of fibre channel fabric names. This
# list of names is used to retrieve other SAN credentials for
# connecting to each SAN fabric (string value)
#fc_fabric_names=<None>

# FC San Lookup Service (string value)
#fc_san_lookup_service=cinder.zonemanager.drivers.brocade.
brcd_fc_san_lookup_service.BrcdFCSanLookupService

[keymgr]

#
# Options defined in cinder.keymgr
#

# The full class name of the key manager API class (string
# value)
#api_class=cinder.keymgr.conf_key_mgr.ConfKeyManager

#
# Options defined in cinder.keymgr.conf_key_mgr
#

# Fixed key returned by key manager, specified in hex (string
# value)
#fixed_key=<None>

[keystone_authtoken]

#
# Options defined in keystoneclient.middleware.auth_token
#

# Prefix to prepend at the beginning of the path. Deprecated,
# use identity_uri. (string value)
#auth_admin_prefix=

# Host providing the admin Identity API endpoint. Deprecated,
# use identity_uri. (string value)
#auth_host=127.0.0.1

# Port of the admin Identity API endpoint. Deprecated, use
# identity_uri. (integer value)
#auth_port=35357

# Protocol of the admin Identity API endpoint (http or https).
# Deprecated, use identity_uri. (string value)
#auth_protocol=https

# Complete public Identity API endpoint (string value)
#auth_uri=<None>
```

```
# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
#identity_uri=<None>

# API version of the admin Identity API endpoint (string
# value)
#auth_version=<None>

# Do not handle authorization requests within the middleware,
# but delegate the authorization decision to downstream WSGI
# components (boolean value)
#delay_auth_decision=false

# Request timeout value for communicating with Identity API
# server. (boolean value)
#http_connect_timeout=<None>

# How many times are we trying to reconnect when communicating
# with Identity API Server. (integer value)
#http_request_max_retries=3

# This option is deprecated and may be removed in a future
# release. Single shared secret with the Keystone
# configuration used for bootstrapping a Keystone
# installation, or otherwise bypassing the normal
# authentication process. This option should not be used, use
# `admin_user` and `admin_password` instead. (string value)
#admin_token=<None>

# Keystone account username (string value)
#admin_user=<None>

# Keystone account password (string value)
#admin_password=<None>

# Keystone service account tenant name to validate user tokens
# (string value)
#admin_tenant_name=admin

# Env key for the swift cache (string value)
#cache=<None>

# Required if Keystone server requires client certificate
# (string value)
#certfile=<None>

# Required if Keystone server requires client certificate
# (string value)
#keyfile=<None>

# A PEM encoded Certificate Authority to use when verifying
# HTTPS connections. Defaults to system CAs. (string value)
#cafile=<None>

# Verify HTTPS connections. (boolean value)
#insecure=false

# Directory used to cache files related to PKI tokens (string
```

```
# value)
#signing_dir=<None>

# Optionally specify a list of memcached server(s) to use for
# caching. If left undefined, tokens will instead be cached
# in-process. (list value)
# Deprecated group/name - [DEFAULT]/memcache_servers
#memcached_servers=<None>

# 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. (integer value)
#token_cache_time=300

# Determines the frequency at which the list of revoked tokens
# is retrieved from the Identity service (in seconds). A high
# number of revocation events combined with a low cache
# duration may significantly reduce performance. (integer
# value)
#revocation_cache_time=10

# (optional) if defined, indicate whether token data should be
# authenticated or authenticated and encrypted. Acceptable
# values are MAC or ENCRYPT. 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. (string value)
#memcache_security_strategy=<None>

# (optional, mandatory if memcache_security_strategy is
# defined) this string is used for key derivation. (string
# value)
#memcache_secret_key=<None>

# (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. (boolean value)
#include_service_catalog=true

# 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. (string
# value)
#enforce_token_bind=permissive

# If true, the revocation list will be checked for cached
# tokens. This requires that PKI tokens are configured on the
# Keystone server. (boolean value)
#check_revocations_for_cached=false

# Hash algorithms to use for hashing PKI tokens. This may be a
# single algorithm or multiple. The algorithms are those
```

```
# supported by Python standard hashlib.new(). The hashes will
# be tried in the order given, so put the preferred one first
# for performance. The result of the first hash will be stored
# in the cache. This will typically be set to multiple values
# only while migrating from a less secure algorithm to a more
# secure one. Once all the old tokens are expired this option
# should be set to a single value for better performance.
# (list value)
#hash_algorithms=md5

[matchmaker_redis]

#
# Options defined in oslo.messaging
#

# Host to locate redis. (string value)
#host=127.0.0.1

# Use this port to connect to redis host. (integer value)
#port=6379

# Password for Redis server (optional). (string value)
#password=<None>

[matchmaker_ring]

#
# Options defined in oslo.messaging
#

# Matchmaker ring file (JSON). (string value)
# Deprecated group/name - [DEFAULT]/matchmaker_ringfile
#ringfile=/etc/oslo/matchmaker_ring.json

[oslo_messaging_amqp]

#
# Options defined in oslo.messaging
#
# NOTE: Options in this group are supported when using oslo.messaging >=1.5.0.

# address prefix used when sending to a specific server
# (string value)
#server_request_prefix=exclusive

# address prefix used when broadcasting to all servers (string
# value)
#broadcast_prefix=broadcast

# address prefix when sending to any server in group (string
# value)
#group_request_prefix=unicast

# Name for the AMQP container (string value)
#container_name=<None>
```

```
# Timeout for inactive connections (in seconds) (integer
# value)
#idle_timeout=0

# Debug: dump AMQP frames to stdout (boolean value)
#trace=false

# CA certificate PEM file for verifying server certificate
# (string value)
#ssl_ca_file=

# Identifying certificate PEM file to present to clients
# (string value)
#ssl_cert_file=

# Private key PEM file used to sign cert_file certificate
# (string value)
#ssl_key_file=

# Password for decrypting ssl_key_file (if encrypted) (string
# value)
#ssl_key_password=<None>

# Accept clients using either SSL or plain TCP (boolean value)
#allow_insecure_clients=false

[profiler]

#
# Options defined in cinder.service
#

# If False fully disable profiling feature. (boolean value)
#profiler_enabled=false

# If False doesn't trace SQL requests. (boolean value)
#trace_sqlalchemy=false

[ssl]

#
# Options defined in cinder.openstack.common.sslutils
#

# CA certificate file to use to verify connecting clients
# (string value)
#ca_file=<None>

# Certificate file to use when starting the server securely
# (string value)
#cert_file=<None>

# Private key file to use when starting the server securely
# (string value)
#key_file=<None>
```


api-paste.ini

Use the `api-paste.ini` file to configure the Block Storage API service.

```
#####
# OpenStack #
#####

[composite:osapi_volume]
use = call:cinder.api.root_app_factory
/: apiversions
/v1: openstack_volume_api_v1
/v2: openstack_volume_api_v2

[composite:openstack_volume_api_v1]
use = call:cinder.api.middleware.auth.pipeline_factory
noauth = request_id faultwrap sizelimit noauth apiv1
keystone = request_id faultwrap sizelimit authtoken keystonecontext apiv1
keystone_nolimit = request_id faultwrap sizelimit authtoken keystonecontext
  apiv1

[composite:openstack_volume_api_v2]
use = call:cinder.api.middleware.auth.pipeline_factory
noauth = request_id faultwrap sizelimit noauth apiv2
keystone = request_id faultwrap sizelimit authtoken keystonecontext apiv2
keystone_nolimit = request_id faultwrap sizelimit authtoken keystonecontext
  apiv2

[filter:request_id]
paste.filter_factory = cinder.openstack.common.middleware.
request_id:RequestIdMiddleware.factory

[filter:faultwrap]
paste.filter_factory = cinder.api.middleware.fault:FaultWrapper.factory

[filter:noauth]
paste.filter_factory = cinder.api.middleware.auth:NoAuthMiddleware.factory

[filter:sizelimit]
paste.filter_factory = cinder.api.middleware.sizelimit:RequestBodySizeLimiter.
factory

[app:apiv1]
paste.app_factory = cinder.api.v1.router:APIRouter.factory

[app:apiv2]
paste.app_factory = cinder.api.v2.router:APIRouter.factory

[pipeline:apiversions]
pipeline = faultwrap osvolumeverSIONapp

[app:osvolumeverSIONapp]
paste.app_factory = cinder.api.versions:Versions.factory

#####
# Shared #
#####
```

```
[filter:keystonecontext]
paste.filter_factory = cinder.api.middleware.auth:CinderKeystoneContext.
factory

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
```

policy.json

The `policy.json` file defines additional access controls that apply to the Block Storage service.

```
{
  "context_is_admin": [{"role:admin"}],
  "admin_or_owner": [{"is_admin:True}], [{"project_id:%(project_id)s"}],
  "default": [{"rule:admin_or_owner"}],

  "admin_api": [{"is_admin:True}],

  "volume:create": [],
  "volume:get_all": [],
  "volume:get_volume_metadata": [],
  "volume:get_volume_admin_metadata": [{"rule:admin_api"}],
  "volume:delete_volume_admin_metadata": [{"rule:admin_api"}],
  "volume:update_volume_admin_metadata": [{"rule:admin_api"}],
  "volume:get_snapshot": [],
  "volume:get_all_snapshots": [],
  "volume:extend": [],
  "volume:update_readonly_flag": [],
  "volume:retype": [],

  "volume_extension:types_manage": [{"rule:admin_api"}],
  "volume_extension:types_extra_specs": [{"rule:admin_api"}],
  "volume_extension:volume_type_encryption": [{"rule:admin_api"}],
  "volume_extension:volume_encryption_metadata": [{"rule:admin_or_owner"}],
  "volume_extension:extended_snapshot_attributes": [],
  "volume_extension:volume_image_metadata": [],

  "volume_extension:quotas:show": [],
  "volume_extension:quotas:update": [{"rule:admin_api"}],
  "volume_extension:quota_classes": [],

  "volume_extension:volume_admin_actions:reset_status":
  [{"rule:admin_api"}],
  "volume_extension:snapshot_admin_actions:reset_status":
  [{"rule:admin_api"}],
  "volume_extension:volume_admin_actions:force_delete":
  [{"rule:admin_api"}],
  "volume_extension:snapshot_admin_actions:force_delete":
  [{"rule:admin_api"}],
  "volume_extension:volume_admin_actions:migrate_volume":
  [{"rule:admin_api"}],
  "volume_extension:volume_admin_actions:migrate_volume_completion":
  [{"rule:admin_api"}],

  "volume_extension:volume_host_attribute": [{"rule:admin_api"}],
```

```
"volume_extension:volume_tenant_attribute": [{"rule:admin_or_owner"}],
"volume_extension:volume_mig_status_attribute": [{"rule:admin_api"}],
"volume_extension:hosts": [{"rule:admin_api"}],
"volume_extension:services": [{"rule:admin_api"}],
"volume:services": [{"rule:admin_api"}],

"volume:create_transfer": [],
"volume:accept_transfer": [],
"volume:delete_transfer": [],
"volume:get_all_transfers": [],

"backup:create" : [],
"backup:delete": [],
"backup:get": [],
"backup:get_all": [],
"backup:restore": [],
"backup:backup-import": [{"rule:admin_api"}],
"backup:backup-export": [{"rule:admin_api"}],

"snapshot_extension:snapshot_actions:update_snapshot_status": []
}
```

rootwrap.conf

The `rootwrap.conf` file defines configuration values used by the `rootwrap` script when the Block Storage service must escalate its privileges to those of the root user.

```
# Configuration for cinder-rootwrap
# This file should be owned by (and only-writeable by) the root user

[DEFAULT]
# List of directories to load filter definitions from (separated by ',').
# These directories MUST all be only writeable by root !
filters_path=/etc/cinder/rootwrap.d,/usr/share/cinder/rootwrap

# List of directories to search executables in, in case filters do not
# explicitly specify a full path (separated by ',')
# If not specified, defaults to system PATH environment variable.
# These directories MUST all be only writeable by root !
exec_dirs=/sbin,/usr/sbin,/bin,/usr/bin

# Enable logging to syslog
# Default value is False
use_syslog=False

# Which syslog facility to use.
# Valid values include auth, authpriv, syslog, local0, local1...
# Default value is 'syslog'
syslog_log_facility=syslog

# Which messages to log.
# INFO means log all usage
# ERROR means only log unsuccessful attempts
syslog_log_level=ERROR
```

Log files used by Block Storage

The corresponding log file of each Block Storage service is stored in the `/var/log/cinder/` directory of the host on which each service runs.

Table 1.33. Log files used by Block Storage services

Log file	Service/interface (for CentOS, Fedora, openSUSE, Red Hat Enterprise Linux, and SUSE Linux Enterprise)	Service/interface (for Ubuntu and Debian)
<code>api.log</code>	<code>openstack-cinder-api</code>	<code>cinder-api</code>
<code>cinder-manage.log</code>	<code>cinder-manage</code>	<code>cinder-manage</code>
<code>scheduler.log</code>	<code>openstack-cinder-scheduler</code>	<code>cinder-scheduler</code>
<code>volume.log</code>	<code>openstack-cinder-volume</code>	<code>cinder-volume</code>

Fibre Channel Zone Manager

The Fibre Channel Zone Manager allows FC SAN Zone/Access control management in conjunction with Fibre Channel block storage. The configuration of Fibre Channel Zone Manager and various zone drivers are described in this section.

Configure Block Storage to use Fibre Channel Zone Manager

If Block Storage is configured to use a Fibre Channel volume driver that supports Zone Manager, update `cinder.conf` to add the following configuration options to enable Fibre Channel Zone Manager.

Make the following changes in the `/etc/cinder/cinder.conf` file.

Table 1.34. Description of configuration options for zoning

Configuration option = Default value	Description
[DEFAULT]	
<code>zoning_mode = none</code>	(StrOpt) FC Zoning mode configured
[fc-zone-manager]	
<code>fc_fabric_names = None</code>	(StrOpt) Comma separated list of fibre channel fabric names. This list of names is used to retrieve other SAN credentials for connecting to each SAN fabric
<code>zoning_policy = initiator-target</code>	(StrOpt) Zoning policy configured by user

To use different Fibre Channel Zone Drivers, use the parameters described in this section.



Note

When multi backend configuration is used, provide the `zoning_mode` configuration option as part of the volume driver configuration where `volume_driver` option is specified.



Note

Default value of `zoning_mode` is `None` and this needs to be changed to `fabric` to allow fabric zoning.

**Note**

`zoning_policy` can be configured as `initiator-target` or `initiator`

Brocade Fibre Channel Zone Driver

Brocade Fibre Channel Zone Driver performs zoning operations via SSH. Configure Brocade Zone Driver and lookup service by specifying the following parameters:

Table 1.35. Description of configuration options for `zoning_manager`

Configuration option = Default value	Description
[fc-zone-manager]	
<code>brcd_sb_connector = cinder.zonemanager.drivers.brocade.brcd_fc_zone_client_cli</code>	(StrOpt) Southbound connector for zoning operation BrcdFCZoneClientCLI
<code>fc_san_lookup_service = cinder.zonemanager.drivers.brocade.brcd_fc_san_lookup_service</code>	(StrOpt) FC San Lookup Service BrcdFCSanLookupService
<code>zone_driver = cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver</code>	(StrOpt) FC Zone Driver responsible for zone management BrcdFCZoneDriver

Configure SAN fabric parameters in the form of fabric groups as described in the example below:

Table 1.36. Description of configuration options for `zoning_fabric`

Configuration option = Default value	Description
[BRCD_FABRIC_EXAMPLE]	
<code>fc_fabric_address =</code>	(StrOpt) Management IP of fabric
<code>fc_fabric_password =</code>	(StrOpt) Password for user
<code>fc_fabric_port = 22</code>	(IntOpt) Connecting port
<code>fc_fabric_user =</code>	(StrOpt) Fabric user ID
<code>principal_switch_wwn = None</code>	(StrOpt) Principal switch WWN of the fabric
<code>zone_activate = True</code>	(BoolOpt) overridden zoning activation state
<code>zone_name_prefix = None</code>	(StrOpt) overridden zone name prefix
<code>zoning_policy = initiator-target</code>	(StrOpt) overridden zoning policy

**Note**

Define a fabric group for each fabric using the fabric names used in `fc_fabric_names` configuration option as group name.

System requirements

Brocade Fibre Channel Zone Driver requires firmware version FOS v6.4 or higher.

As a best practice for zone management, use a user account with `zoneadmin` role. Users with `admin` role (including the default `admin` user account) are limited to a maximum of two concurrent SSH sessions.

For information about how to manage Brocade Fibre Channel switches, see the Brocade Fabric OS user documentation.

Additional options

These options can also be set in the `cinder.conf` file.

Table 1.37. Description of configuration options for `auth_token`

Configuration option = Default value	Description
[DEFAULT]	
<code>memcached_servers = None</code>	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
<code>admin_password = None</code>	(StrOpt) Keystone account password
<code>admin_tenant_name = admin</code>	(StrOpt) Keystone service account tenant name to validate user tokens
<code>admin_token = None</code>	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
<code>admin_user = None</code>	(StrOpt) Keystone account username
<code>auth_admin_prefix =</code>	(StrOpt) Prefix to prepend at the beginning of the path
<code>auth_host = 127.0.0.1</code>	(StrOpt) Host providing the admin Identity API endpoint
<code>auth_port = 35357</code>	(IntOpt) Port of the admin Identity API endpoint
<code>auth_protocol = https</code>	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
<code>auth_uri = None</code>	(StrOpt) Complete public Identity API endpoint
<code>auth_version = None</code>	(StrOpt) API version of the admin Identity API endpoint
<code>cache = None</code>	(StrOpt) Env key for the swift cache
<code>cafile = None</code>	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
<code>certfile = None</code>	(StrOpt) Required if Keystone server requires client certificate
<code>delay_auth_decision = False</code>	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
<code>enforce_token_bind = permissive</code>	(StrOpt) 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.
<code>http_connect_timeout = None</code>	(BoolOpt) Request timeout value for communicating with Identity API server.
<code>http_request_max_retries = 3</code>	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
<code>include_service_catalog = True</code>	(BoolOpt) (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.
<code>insecure = False</code>	(BoolOpt) Verify HTTPS connections.
<code>keyfile = None</code>	(StrOpt) Required if Keystone server requires client certificate

Configuration option = Default value	Description
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. 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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

Table 1.38. Description of configuration options for connection

Configuration option = Default value	Description
[database]	
connection = sqlite:///state_path/sqlite_db	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings

Table 1.39. Description of configuration options for huawei

Configuration option = Default value	Description
[DEFAULT]	
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml	(StrOpt) config data for cinder huawei plugin

Table 1.40. Description of configuration options for nas

Configuration option = Default value	Description
[DEFAULT]	
nas_ip =	(StrOpt) IP address or Hostname of NAS system.
nas_login = admin	(StrOpt) User name to connect to NAS system.
nas_password =	(StrOpt) Password to connect to NAS system.
nas_private_key =	(StrOpt) Filename of private key to use for SSH authentication.
nas_ssh_port = 22	(IntOpt) SSH port to use to connect to NAS system.

Table 1.41. Description of configuration options for hpmsa

Configuration option = Default value	Description
[DEFAULT]	
msa_vdisk = OpenStack	(StrOpt) The VDisk to use for volume creation.

Table 1.42. Description of configuration options for database

Configuration option = Default value	Description
[DEFAULT]	
db_backend = sqlalchemy	(StrOpt) The backend to use for db
db_driver = cinder.db	(StrOpt) driver to use for database access

Table 1.43. Description of configuration options for keymgr

Configuration option = Default value	Description
[keymgr]	
api_class = cinder.keymgr.conf_key_mgr.ConfKeyManager	(StrOpt) The full class name of the key manager API class
fixed_key = None	(StrOpt) Fixed key returned by key manager, specified in hex

Table 1.44. Description of configuration options for storage

Configuration option = Default value	Description
[DEFAULT]	
allocated_capacity_weight_multiplier = -1.0	(FloatOpt) Multiplier used for weighing volume capacity. Negative numbers mean to stack vs spread.
capacity_weight_multiplier = 1.0	(FloatOpt) Multiplier used for weighing volume capacity. Negative numbers mean to stack vs spread.
enabled_backends = None	(ListOpt) A list of backend names to use. These backend names should be backed by a unique [CONFIG] group with its options
iscsi_helper = tgtadm	(StrOpt) iscsi target user-land tool to use
iscsi_iotype = fileio	(StrOpt) Sets the behavior of the iSCSI target to either perform blockio or fileio optionally, auto can be set and Cinder will autodetect type of backing device
iscsi_ip_address = \$my_ip	(StrOpt) The IP address that the iSCSI daemon is listening on
iscsi_num_targets = 100	(IntOpt) The maximum number of iscsi target ids per host
iscsi_port = 3260	(IntOpt) The port that the iSCSI daemon is listening on
iscsi_target_prefix = iqn.2010-10.org.openstack:	(StrOpt) prefix for iscsi volumes
iser_helper = tgtadm	(StrOpt) iser target user-land tool to use
iser_ip_address = \$my_ip	(StrOpt) The IP address that the iSER daemon is listening on
iser_num_targets = 100	(IntOpt) The maximum number of iser target ids per host
iser_port = 3260	(IntOpt) The port that the iSER daemon is listening on
iser_target_prefix = iqn.2010-10.org.iser.openstack:	(StrOpt) prefix for iser volumes
max_gigabytes = 10000	(IntOpt) This configure option has been deprecated along with the SimpleScheduler. New scheduler is able to gather capacity information for each host, thus setting the maximum number of volume gigabytes for host is no longer needed. It's safe to remove this configure from cinder.conf.

Configuration option = Default value	Description
migration_create_volume_timeout_secs = 300	(IntOpt) Timeout for creating the volume to migrate to when performing volume migration (seconds)
num_iscsi_scan_tries = 3	(IntOpt) The maximum number of times to rescan iSER target to find volume
num_volume_device_scan_tries = 3	(IntOpt) The maximum number of times to rescan targets to find volume
volume_backend_name = None	(StrOpt) The backend name for a given driver implementation
volume_clear = zero	(StrOpt) Method used to wipe old volumes (valid options are: none, zero, shred)
volume_clear_ionice = None	(StrOpt) The flag to pass to ionice to alter the i/o priority of the process used to zero a volume after deletion, for example "-c3" for idle only priority.
volume_clear_size = 0	(IntOpt) Size in MiB to wipe at start of old volumes. 0 => all
volume_dd_blocksize = 1M	(StrOpt) The default block size used when copying/clearing volumes
volume_driver = cinder.volume.drivers.lvm.LVMISCSIDriver	(StrOpt) Driver to use for volume creation
volume_manager = cinder.volume.manager.VolumeManager	(StrOpt) full class name for the Manager for volume
volume_service_inithost_offload = False	(BoolOpt) Offload pending volume delete during volume service startup
volume_usage_audit_period = month	(StrOpt) time period to generate volume usages for. Time period must be hour, day, month or year
volumes_dir = \$state_path/volumes	(StrOpt) Volume configuration file storage directory
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = 5	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool

Table 1.45. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
allowed_rpc_exception_modules = oslo.messaging.exceptions, nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
control_exchange = openstack	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
default_timeout = 20	(IntOpt) Default Time Out For CLI operations in minutes
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Configuration option = Default value	Description
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = notifications	(ListOpt) AMQP topic used for OpenStack notifications.
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.
publish_errors = False	(BoolOpt) Publish error events
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
qpid_password =	(StrOpt) Password for Qpid connection.
qpid_port = 5672	(IntOpt) Qpid broker port.
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for Qpid connection.
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
rabbit_password = guest	(StrOpt) The RabbitMQ password.
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
rabbit_userid = guest	(StrOpt) The RabbitMQ userid.

Configuration option = Default value	Description
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
rpc_backend = rabbit	(StrOpt) The messaging driver to use, defaults to rabbit. Other drivers include qpid and zmq.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC greenthread pool.
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1.
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets.
rpc_zmq_matchmaker = oslo.messaging_drivers.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver.
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port.
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.
volume_topic = cinder-volume	(StrOpt) the topic volume nodes listen on
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).

Table 1.46. Description of configuration options for san-solaris

Configuration option = Default value	Description
[DEFAULT]	
san_zfs_volume_base = rpool/	(StrOpt) The ZFS path under which to create zvols for volumes.

Table 1.47. Description of configuration options for rootwrap

Configuration option = Default value	Description
[DEFAULT]	
filters_path = /etc/cinder/rootwrap.d,/usr/share/cinder/rootwrap	List of directories to load filter definitions from (separated by ';'). These directories MUST all be only writeable by root !
exec_dirs = /sbin,/usr/sbin,/bin,/usr/bin	List of directories to search executables in, in case filters do not explicitly specify a full path (separated by ';') If not specified, defaults to system PATH environment variable. These directories MUST all be only writeable by root !
use_syslog = False	Enable logging to syslog Default value is False
syslog_log_facility = syslog	Which syslog facility to use. Valid values include auth, authpriv, syslog, local0, local1... Default value is 'syslog'
syslog_log_level = ERROR	Which messages to log. INFO means log all usage ERROR means only log unsuccessful attempts

Table 1.48. Description of configuration options for ssl

Configuration option = Default value	Description
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 1.49. Description of configuration options for images

Configuration option = Default value	Description
[DEFAULT]	
allowed_direct_url_schemes =	(ListOpt) A list of url schemes that can be downloaded directly via the direct_url. Currently supported schemes: [file].
glance_api_insecure = False	(BoolOpt) Allow to perform insecure SSL (https) requests to glance
glance_api_servers = \$glance_host:\$glance_port	(ListOpt) A list of the glance api servers available to cinder ([hostname ip]:port)
glance_api_ssl_compression = False	(BoolOpt) Whether to attempt to negotiate SSL layer compression when using SSL (https) requests. Set to False to disable SSL layer compression. In some cases disabling this may improve data throughput, eg when high network bandwidth is available and you are using already compressed image formats such as qcow2 .
glance_api_version = 1	(IntOpt) Version of the glance api to use
glance_host = \$my_ip	(StrOpt) default glance hostname or ip
glance_num_retries = 0	(IntOpt) Number retries when downloading an image from glance
glance_port = 9292	(IntOpt) default glance port
glance_request_timeout = None	(IntOpt) http/https timeout value for glance operations. If no value (None) is supplied here, the glanceclient default value is used.
image_conversion_dir = \$state_path/conversion	(StrOpt) Directory used for temporary storage during image conversion
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
use_multipath_for_image_xfer = False	(BoolOpt) Do we attach/detach volumes in cinder using multipath for volume to image and image to volume transfers?

Table 1.50. Description of configuration options for emc

Configuration option = Default value	Description
[DEFAULT]	
cinder_emc_config_file = /etc/cinder/cinder_emc_config.xml	(StrOpt) use this file for cinder emc plugin config data
max_luns_per_storage_group = 256	(IntOpt) Default max number of LUNs in a storage group
naviseccli_path =	(StrOpt) Naviseccli Path
storage_vnx_pool_name = None	(StrOpt) ISCSI pool name

Table 1.51. Description of configuration options for backups

Configuration option = Default value	Description
[DEFAULT]	
backup_api_class = cinder.backup.api.API	(StrOpt) The full class name of the volume backup API class
backup_compression_algorithm = zlib	(StrOpt) Compression algorithm (None to disable)
backup_driver = cinder.backup.drivers.swift	(StrOpt) Driver to use for backups.
backup_manager = cinder.backup.manager.BackupManager	(StrOpt) full class name for the Manager for volume backup
backup_metadata_version = 1	(IntOpt) Backup metadata version to be used when backing up volume metadata. If this number is bumped, make sure the service doing the restore supports the new version.
backup_name_template = backup-%s	(StrOpt) Template string to be used to generate backup names
backup_topic = cinder-backup	(StrOpt) the topic volume backup nodes listen on
snapshot_name_template = snapshot-%s	(StrOpt) Template string to be used to generate snapshot names
snapshot_same_host = True	(BoolOpt) Create volume from snapshot at the host where snapshot resides

Table 1.52. Description of configuration options for hp3par

Configuration option = Default value	Description
[DEFAULT]	
hp3par_api_url =	(StrOpt) 3PAR WSAPI Server Url like https://<3par ip>:8080/api/v1
hp3par_cpg = OpenStack	(StrOpt) The CPG to use for volume creation
hp3par_cpg_snap =	(StrOpt) The CPG to use for Snapshots for volumes. If empty hp3par_cpg will be used
hp3par_debug = False	(BoolOpt) Enable HTTP debugging to 3PAR
hp3par_iscsi_ips =	(ListOpt) List of target iSCSI addresses to use.
hp3par_password =	(StrOpt) 3PAR Super user password
hp3par_snapshot_expiration =	(StrOpt) The time in hours when a snapshot expires and is deleted. This must be larger than expiration
hp3par_snapshot_retention =	(StrOpt) The time in hours to retain a snapshot. You can't delete it before this expires.
hp3par_username =	(StrOpt) 3PAR Super user username

Table 1.53. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
api_paste_config = api-paste.ini	(StrOpt) File name for the paste.deploy config for cinder-api
api_rate_limit = True	(BoolOpt) whether to rate limit the api
backdoor_port = None	(StrOpt) 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

Configuration option = Default value	Description
	range of port numbers. The chosen port is displayed in the service's log file.
enable_v1_api = True	(BoolOpt) Deploy v1 of the Cinder API.
enable_v2_api = True	(BoolOpt) Deploy v2 of the Cinder API.
extra_capabilities = {}	(StrOpt) User defined capabilities, a JSON formatted string specifying key/value pairs.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
osapi_max_limit = 1000	(IntOpt) the maximum number of items returned in a single response from a collection resource
osapi_max_request_body_size = 114688	(IntOpt) Max size for body of a request
osapi_volume_base_URL = None	(StrOpt) Base URL that will be presented to users in links to the OpenStack Volume API
osapi_volume_ext_list =	(ListOpt) Specify list of extensions to load when using osapi_volume_extension option with cinder.api.contrib.select_extensions
osapi_volume_extension = ['cinder.api.contrib.standard_extensions']	(MultiStrOpt) osapi volume extension to load
osapi_volume_listen = 0.0.0.0	(StrOpt) IP address for OpenStack Volume API to listen
osapi_volume_listen_port = 8776	(IntOpt) port for os volume api to listen
osapi_volume_workers = None	(IntOpt) Number of workers for OpenStack Volume API service
transfer_api_class = cinder.transfer.api.API	(StrOpt) The full class name of the volume transfer API class
volume_api_class = cinder.volume.api.API	(StrOpt) The full class name of the volume API class to use
volume_name_template = volume-%s	(StrOpt) Template string to be used to generate volume names
volume_transfer_key_length = 16	(IntOpt) The number of characters in the autogenerated auth key.
volume_transfer_salt_length = 8	(IntOpt) The number of characters in the salt.

Table 1.54. Description of configuration options for hplefthand

Configuration option = Default value	Description
[DEFAULT]	
hplefthand_api_url = None	(StrOpt) HP LeftHand WSAPI Server Url like https://<LeftHand ip>:8081/lhos
hplefthand_clustername = None	(StrOpt) HP LeftHand cluster name
hplefthand_debug = False	(BoolOpt) Enable HTTP debugging to LeftHand
hplefthand_iscsi_chap_enabled = False	(BoolOpt) Configure CHAP authentication for iSCSI connections (Default: Disabled)
hplefthand_password = None	(StrOpt) HP LeftHand Super user password
hplefthand_username = None	(StrOpt) HP LeftHand Super user username

Table 1.55. Description of configuration options for scalability

Configuration option = Default value	Description
[DEFAULT]	
scality_sofs_config = None	(StrOpt) Path or URL to Scality SOFS configuration file
scality_sofs_mount_point = \$state_path/scality	(StrOpt) Base dir where Scality SOFS shall be mounted

Configuration option = Default value	Description
scality_sofs_volume_dir = cinder/volumes	(StrOpt) Path from Scality SOFS root to volume dir

Table 1.56. Description of configuration options for block-device

Configuration option = Default value	Description
[DEFAULT]	
available_devices =	(ListOpt) List of all available devices

Table 1.57. Description of configuration options for compute

Configuration option = Default value	Description
[DEFAULT]	
nova_api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to nova
nova_ca_certificates_file = None	(StrOpt) Location of ca certificates file to use for nova client requests.
nova_catalog_admin_info = compute:nova:adminURL	(StrOpt) Same as nova_catalog_info, but for admin endpoint.
nova_catalog_info = compute:nova:publicURL	(StrOpt) Info to match when looking for nova in the service catalog. Format is : separated values of the form: <service_type>:<service_name>:<endpoint_type>
nova_endpoint_admin_template = None	(StrOpt) Same as nova_endpoint_template, but for admin endpoint.
nova_endpoint_template = None	(StrOpt) Override service catalog lookup with template for nova endpoint e.g. http://localhost:8774/v2/%(project_id)s
os_region_name = None	(StrOpt) region name of this node

Table 1.58. Description of configuration options for san

Configuration option = Default value	Description
[DEFAULT]	
san_clustername =	(StrOpt) Cluster name to use for creating volumes
san_ip =	(StrOpt) IP address of SAN controller
san_is_local = False	(BoolOpt) Execute commands locally instead of over SSH; use if the volume service is running on the SAN device
san_login = admin	(StrOpt) Username for SAN controller
san_password =	(StrOpt) Password for SAN controller
san_private_key =	(StrOpt) Filename of private key to use for SSH authentication
san_ssh_port = 22	(IntOpt) SSH port to use with SAN
san_thin_provision = True	(BoolOpt) Use thin provisioning for SAN volumes?
ssh_conn_timeout = 30	(IntOpt) SSH connection timeout in seconds
ssh_max_pool_conn = 5	(IntOpt) Maximum ssh connections in the pool
ssh_min_pool_conn = 1	(IntOpt) Minimum ssh connections in the pool

Table 1.59. Description of configuration options for zones

Configuration option = Default value	Description
[DEFAULT]	
cloned_volume_same_az = True	(BoolOpt) Ensure that the new volumes are the same AZ as snapshot or source volume

Table 1.60. Description of configuration options for auth

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = noauth	(StrOpt) The strategy to use for auth. Supports noauth, keystone, and deprecated.

Table 1.61. Description of configuration options for scheduler

Configuration option = Default value	Description
[DEFAULT]	
scheduler_default_filters = AvailabilityZoneFilter, CapacityFilter, CapabilitiesFilter	(ListOpt) Which filter class names to use for filtering hosts when not specified in the request.
scheduler_default_weighers = CapacityWeigher	(ListOpt) Which weigher class names to use for weighing hosts.
scheduler_driver = cinder.scheduler.filter_scheduler.FilterScheduler	(StrOpt) Default scheduler driver to use
scheduler_host_manager = cinder.scheduler.host_manager.HostManager	(StrOpt) The scheduler host manager class to use
scheduler_json_config_location =	(StrOpt) Absolute path to scheduler configuration JSON file.
scheduler_manager = cinder.scheduler.manager.SchedulerManager	(StrOpt) full class name for the Manager for scheduler
scheduler_max_attempts = 3	(IntOpt) Maximum number of attempts to schedule an volume
scheduler_topic = cinder-scheduler	(StrOpt) the topic scheduler nodes listen on

Table 1.62. Description of configuration options for quota

Configuration option = Default value	Description
[DEFAULT]	
max_age = 0	(IntOpt) number of seconds between subsequent usage refreshes
quota_driver = cinder.quota.DbQuotaDriver	(StrOpt) default driver to use for quota checks
quota_gigabytes = 1000	(IntOpt) number of volume gigabytes (snapshots are also included) allowed per project
quota_snapshots = 10	(IntOpt) number of volume snapshots allowed per project
quota_volumes = 10	(IntOpt) number of volumes allowed per project
reservation_expire = 86400	(IntOpt) number of seconds until a reservation expires
use_default_quota_class = True	(BoolOpt) whether to use default quota class for default quota

Table 1.63. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
compute_api_class = cinder.compute.nova.API	(StrOpt) The full class name of the compute API class to use
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
default_availability_zone = None	(StrOpt) default availability zone to use when creating a new volume. If this is not set then we use the value from the storage_availability_zone option as the default availability_zone for new volumes.

Configuration option = Default value	Description
default_log_levels = amqp=WARN, amqplib=WARN, boto=WARN, qpids=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN	(ListOpt) List of logger=LEVEL pairs
default_volume_type = None	(StrOpt) default volume type to use
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
enable_new_services = True	(BoolOpt) Services to be added to the available pool on create
fatal_deprecations = False	(BoolOpt) Make deprecations fatal
fatal_exception_format_errors = False	(BoolOpt) make exception message format errors fatal
host = oslo	(StrOpt) Name of this node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address.
iet_conf = /etc/iet/ietd.conf	(StrOpt) IET configuration file
lio_initiator_iqns =	(StrOpt) Comma-separated list of initiator IQNs allowed to connect to the iSCSI target. (From Nova compute nodes.)
lock_path = None	(StrOpt) Directory to use for lock files. Default to a temp directory
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%request_id)s %(user_identity)s %(instance)s %(message)s	(StrOpt) Format string to use for log messages with context
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) Data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) Format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) Prefix each line of exception output with this format
monkey_patch = False	(BoolOpt) Enable monkey patching
monkey_patch_modules =	(ListOpt) List of modules/decorators to monkey patch
my_ip = 10.0.0.1	(StrOpt) ip address of this host
no_snapshot_gb_quota = False	(BoolOpt) Whether snapshots count against GigaByte quota
num_shell_tries = 3	(IntOpt) number of times to attempt to run flakey shell commands

Configuration option = Default value	Description
periodic_fuzzy_delay = 60	(IntOpt) range of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = 60	(IntOpt) seconds between running periodic tasks
policy_default_rule = default	(StrOpt) Rule checked when requested rule is not found
policy_file = policy.json	(StrOpt) JSON file representing policy
report_interval = 10	(IntOpt) seconds between nodes reporting state to datastore
reserved_percentage = 0	(IntOpt) The percentage of backend capacity is reserved
rootwrap_config = /etc/cinder/rootwrap.conf	(StrOpt) Path to the rootwrap configuration file to use for running commands as root
run_external_periodic_tasks = True	(BoolOpt) Some periodic tasks can be run in a separate process. Should we run them here?
service_down_time = 60	(IntOpt) maximum time since last check-in for up service
sqlite_db = cinder.sqlite	(StrOpt) the filename to use with sqlite
sqlite_synchronous = True	(BoolOpt) If true, use synchronous mode for sqlite
ssl_ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
ssl_cert_file = None	(StrOpt) Certificate file to use when starting the server securely
ssl_key_file = None	(StrOpt) Private key file to use when starting the server securely
state_path = /var/lib/cinder	(StrOpt) Top-level directory for maintaining cinder's state
storage_availability_zone = nova	(StrOpt) availability zone of this node
syslog_log_facility = LOG_USER	(StrOpt) Syslog facility to receive log lines
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
until_refresh = 0	(IntOpt) count of reservations until usage is refreshed
use_forwarded_for = False	(BoolOpt) Treat X-Forwarded-For as the canonical remote address. Only enable this if you have a sanitizing proxy.
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and then will be changed in J to honor RFC5424
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).
[database]	
idle_timeout = 3600	(IntOpt) timeout before idle sql connections are reaped
retry_interval = 10	(IntOpt) interval between retries of opening a sql connection
use_tpool = False	(BoolOpt) Enable the experimental use of thread pooling for all DB API calls

New, updated and deprecated options for cinder

Table 1.64. New options

Option = default value	(Type) Help string
allocated_capacity_weight_multiplier = -1.0	(FloatOpt) Multiplier used for weighing volume capacity. Negative numbers mean to stack vs spread.
backup_metadata_version = 1	(IntOpt) Backup metadata version to be used when backing up volume metadata. If this number is bumped, make sure the service doing the restore supports the new version.
cinder_emc_config_file = /etc/cinder/cinder_emc_config.xml	(StrOpt) use this file for cinder emc plugin config data
default_timeout = 20	(IntOpt) Default Time Out For CLI operations in minutes
extra_capabilities = {}	(StrOpt) User defined capabilities, a JSON formatted string specifying key/value pairs.
gpfs_storage_pool = None	(StrOpt) Specifies the storage pool that volumes are assigned to. By default, the system storage pool is used.
hplefthand_api_url = None	(StrOpt) HP LeftHand WSAPI Server Url like https://<LeftHand ip>:8081/lhos
hplefthand_clustername = None	(StrOpt) HP LeftHand cluster name
hplefthand_debug = False	(BoolOpt) Enable HTTP debugging to LeftHand
hplefthand_iscsi_chap_enabled = False	(BoolOpt) Configure CHAP authentication for iSCSI connections (Default: Disabled)
hplefthand_password = None	(StrOpt) HP LeftHand Super user password
hplefthand_username = None	(StrOpt) HP LeftHand Super user username
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
max_luns_per_storage_group = 256	(IntOpt) Default max number of LUNs in a storage group
msa_vdisk = OpenStack	(StrOpt) The VDisk to use for volume creation.
nas_ip =	(StrOpt) IP address or Hostname of NAS system.
nas_login = admin	(StrOpt) User name to connect to NAS system.
nas_password =	(StrOpt) Password to connect to NAS system.
nas_private_key =	(StrOpt) Filename of private key to use for SSH authentication.
nas_ssh_port = 22	(IntOpt) SSH port to use to connect to NAS system.
naviseccli_path =	(StrOpt) Naviseccli Path
netapp_controller_ips = None	(StrOpt) This option is only utilized when the storage family is configured to eseries. This option is used to restrict provisioning to the specified controllers. Specify the value of this option to be a comma separated list of controller hostnames or IP addresses to be used for provisioning.
netapp_copyoffload_tool_path = None	(StrOpt) This option specifies the path of the NetApp copy offload tool binary. Ensure that the binary has execute

Option = default value	(Type) Help string
	permissions set which allow the effective user of the cinder-volume process to execute the file.
netapp_sa_password = None	(StrOpt) Password for the NetApp E-Series storage array.
netapp_storage_pools = None	(StrOpt) This option is used to restrict provisioning to the specified storage pools. Only dynamic disk pools are currently supported. Specify the value of this option to be a comma separated list of disk pool names to be used for provisioning.
netapp_webservice_path = /devmgr/v2	(StrOpt) This option is used to specify the path to the E-Series proxy application on a proxy server. The value is combined with the value of the netapp_transport_type, netapp_server_hostname, and netapp_server_port options to create the URL used by the driver to connect to the proxy application.
nexenta_nms_cache_volroot = True	(BoolOpt) If set True cache NexentaStor appliance volroot option value.
nexenta_rrmgr_compression = 0	(IntOpt) Enable stream compression, level 1..9. 1 - gives best speed; 9 - gives best compression.
nexenta_rrmgr_connections = 2	(IntOpt) Number of TCP connections.
nexenta_rrmgr_tcp_buf_size = 4096	(IntOpt) TCP Buffer size in KiloBytes.
osapi_volume_workers = None	(IntOpt) Number of workers for OpenStack Volume API service
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
storage_vnx_pool_name = None	(StrOpt) iSCSI pool name
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
vmware_api_retry_count = 10	(IntOpt) Number of times VMware ESX/VC server API must be retried upon connection related issues.
vmware_host_ip = None	(StrOpt) IP address for connecting to VMware ESX/VC server.
vmware_host_password = None	(StrOpt) Password for authenticating with VMware ESX/VC server.
vmware_host_username = None	(StrOpt) Username for authenticating with VMware ESX/VC server.
vmware_host_version = None	(StrOpt) Optional string specifying the VMware VC server version. The driver attempts to retrieve the version from VMware VC server. Set this configuration only if you want to override the VC server version.
vmware_image_transfer_timeout_secs = 7200	(IntOpt) Timeout in seconds for VMDK volume transfer between Cinder and Glance.
vmware_max_objects_retrieval = 100	(IntOpt) Max number of objects to be retrieved per batch. Query results will be obtained in batches from the server and not in one shot. Server may still limit the count to something less than the configured value.
vmware_task_poll_interval = 5	(IntOpt) The interval (in seconds) for polling remote tasks invoked on VMware ESX/VC server.

Option = default value	(Type) Help string
vmware_volume_folder = cinder-volumes	(StrOpt) Name for the folder in the VC datacenter that will contain cinder volumes.
vmware_wsdl_location = None	(StrOpt) Optional VIM service WSDL Location e.g http://<server>/vimService.wsdl. Optional over-ride to default location for bug work-arounds.
volume_clear_ionice = None	(StrOpt) The flag to pass to ionice to alter the i/o priority of the process used to zero a volume after deletion, for example "-c3" for idle only priority.
xiv_chap = disabled	(StrOpt) CHAP authentication mode, effective only for iscsi (disabled enabled)
zoning_mode = none	(StrOpt) FC Zoning mode configured
BRCD_FABRIC_EXAMPLE/fc_fabric_address =	(StrOpt) Management IP of fabric
BRCD_FABRIC_EXAMPLE/fc_fabric_password =	(StrOpt) Password for user
BRCD_FABRIC_EXAMPLE/fc_fabric_port = 22	(IntOpt) Connecting port
BRCD_FABRIC_EXAMPLE/fc_fabric_user =	(StrOpt) Fabric user ID
BRCD_FABRIC_EXAMPLE/principal_switch_wwn = None	(StrOpt) Principal switch WWN of the fabric
BRCD_FABRIC_EXAMPLE/zone_activate = True	(BoolOpt) overridden zoning activation state
BRCD_FABRIC_EXAMPLE/zone_name_prefix = None	(StrOpt) overridden zone name prefix
BRCD_FABRIC_EXAMPLE/zoning_policy = initiator-target	(StrOpt) overridden zoning policy
fc-zone-manager/brcd_sb_connector = cinder.zonemanager.drivers.brocade.brcd_fc_zone_client_cli.BrcdFCZoneClientCLI	(StrOpt) Southbound connector for zoning operation
fc-zone-manager/fc_fabric_names = None	(StrOpt) Comma separated list of fibre channel fabric names. This list of names is used to retrieve other SAN credentials for connecting to each SAN fabric
fc-zone-manager/fc_san_lookup_service = cinder.zonemanager.drivers.brocade.brcd_fc_san_lookup_service.BrcdFCSanLookupService	(StrOpt) FC San Lookup Service
fc-zone-manager/zone_driver = cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver.BrcdFCZoneDriver	(StrOpt) FC Zone Driver responsible for zone management
fc-zone-manager/zoning_policy = initiator-target	(StrOpt) Zoning policy configured by user
matchmaker_ring/ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).
ssl/ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
ssl/cert_file = None	(StrOpt) Certificate file to use when starting the server securely
ssl/key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 1.65. New default values

Option	Previous default value	New default value
default_log_levels	amqpplib=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, eventlet.wsgi.server=WARN	amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s
rpc_zmq_matchmaker	cinder.openstack.common.rpc.matchmaker.MatchMakerLocalhost	MatchMakerLocalhost
sf_account_prefix	ol.local.net	None

Option	Previous default value	New default value
state_path	\$pybasedir	/var/lib/cinder

Table 1.66. Deprecated options

Deprecated option	New Option
pybasedir	state_path
matchmaker_ringfile	matchmaker_ring/ringfile
log_config	log_config_append
rpc_notifier2/topics	notification_topics
db_backend	database/backend

2. Compute

Table of Contents

Overview of nova.conf	140
Configure logging	141
Configure authentication and authorization	141
Configure resize	142
Database configuration	142
Configure the Oslo RPC messaging system	142
Configure the Compute API	146
Configure the EC2 API	148
Fibre Channel support in Compute	148
Hypervisors	148
Scheduling	180
Cells	195
Conductor	199
Example nova.conf configuration files	199
Compute log files	204
Compute sample configuration files	205
New, updated and deprecated options for nova	244

The OpenStack Compute service is a cloud computing fabric controller, which is the main part of an IaaS system. You can use OpenStack Compute to host and manage cloud computing systems. This section describes the OpenStack Compute configuration options.

To configure your Compute installation, you must define configuration options in these files:

- `nova.conf`. Contains most of the Compute configuration options. Resides in the `/etc/nova` directory.
- `api-paste.ini`. Defines Compute limits. Resides in the `/etc/nova` directory.
- Related Image Service and Identity service management configuration files.

Overview of nova.conf

The `nova.conf` configuration file is an [INI file format](#) as explained in [the section called "Configuration file format" \[17\]](#).

You can use a particular configuration option file by using the `option(nova.conf)` parameter when you run one of the `nova-*` services. This parameter inserts configuration option definitions from the specified configuration file name, which might be useful for debugging or performance tuning.

For a list of configuration options, see the tables in this guide.

To learn more about the `nova.conf` configuration file, review the general purpose configuration options documented in [Table 2.19, “Description of configuration options for common” \[210\]](#).



Important

Do not specify quotes around Nova options.

Sections

Configuration options are grouped by section. The Compute configuration file supports the following sections:

[DEFAULT]	Contains most configuration options. If the documentation for a configuration option does not specify its section, assume that it appears in this section.
[cells]	Configures cells functionality. For details, see the Cells section (../config-reference/content/section_compute-cells.html).
[baremetal]	Configures the baremetal hypervisor driver.
[conductor]	Configures the <code>nova-conductor</code> service.
[trusted_computing]	Configures the trusted computing pools functionality and how to connect to a remote attestation service.

Configure logging

You can use `nova.conf` file to configure where Compute logs events, the level of logging, and log formats.

To customize log formats for OpenStack Compute, use the configuration option settings documented in [Table 2.35, “Description of configuration options for logging” \[219\]](#).

Configure authentication and authorization

There are different methods of authentication for the OpenStack Compute project, including no authentication. The preferred system is the OpenStack Identity service, code-named Keystone.

To customize authorization settings for Compute, use the configuration options documented in [Table 2.13, “Description of configuration options for authentication” \[206\]](#).

To customize certificate authority settings for Compute, use the configuration options documented in [Table 2.17, “Description of configuration options for ca” \[209\]](#).

To customize Compute and the Identity service to use LDAP as a backend, refer to the configuration options documented in [Table 2.32, “Description of configuration options for ldap” \[217\]](#).

Configure resize

Resize (or Server resize) is the ability to change the flavor of a server, thus allowing it to upscale or downscale according to user needs. For this feature to work properly, you might need to configure some underlying virt layers.

KVM

Resize on KVM is implemented currently by transferring the images between compute nodes over ssh. For KVM you need hostnames to resolve properly and passwordless ssh access between your compute hosts. Direct access from one compute host to another is needed to copy the VM file across.

Cloud end users can find out how to resize a server by reading the [OpenStack End User Guide](#).

XenServer

To get resize to work with XenServer (and XCP), you need to establish a root trust between all hypervisor nodes and provide an `/image` mount point to your hypervisors `dom0`.

Database configuration

You can configure OpenStack Compute to use any SQLAlchemy-compatible database. The database name is `nova`. The `nova-conductor` service is the only service that writes to the database. The other Compute services access the database through the `nova-conductor` service.

To ensure that the database schema is current, run the following command:

```
# nova-manage db sync
```

If `nova-conductor` is not used, entries to the database are mostly written by the `nova-scheduler` service, although all services must be able to update entries in the database.

In either case, use the configuration option settings documented in [Table 2.24, "Description of configuration options for db" \[213\]](#) to configure the connection string for the `nova` database.

Configure the Oslo RPC messaging system

OpenStack projects use AMQP, an open standard for messaging middleware. OpenStack services that run on multiple servers to talk to each other. OpenStack Oslo RPC supports three implementations of AMQP: RabbitMQ, Qpid, and ZeroMQ.

Configure RabbitMQ

OpenStack Oslo RPC uses RabbitMQ by default. Use these options to configure the RabbitMQ message system. The `rpc_backend` option is not required as long as RabbitMQ is the default messaging system. However, if it is included the configuration, you must set it to `nova.openstack.common.rpc.impl_kombu`.

```
rpc_backend=nova.openstack.common.rpc.impl_kombu
```

You can use these additional options to configure the RabbitMQ messaging system. You can configure messaging communication for different installation scenarios, tune retries for RabbitMQ, and define the size of the RPC thread pool. To monitor notifications through RabbitMQ, you must set the `notification_driver` option to `nova.notifier.rabbit_notifier` in the `nova.conf` file. The default for sending usage data is sixty seconds plus a random number of seconds from zero to sixty.

Table 2.1. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
<code>rabbit_ha_queues = False</code>	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
<code>rabbit_host = localhost</code>	(StrOpt) The RabbitMQ broker address where a single node is used.
<code>rabbit_hosts = \$rabbit_host:\$rabbit_port</code>	(ListOpt) RabbitMQ HA cluster host:port pairs.
<code>rabbit_login_method = AMQPPLAIN</code>	(StrOpt) the RabbitMQ login method
<code>rabbit_max_retries = 0</code>	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
<code>rabbit_password = guest</code>	(StrOpt) The RabbitMQ password.
<code>rabbit_port = 5672</code>	(IntOpt) The RabbitMQ broker port where a single node is used.
<code>rabbit_retry_backoff = 2</code>	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
<code>rabbit_retry_interval = 1</code>	(IntOpt) How frequently to retry connecting with RabbitMQ.
<code>rabbit_use_ssl = False</code>	(BoolOpt) Connect over SSL for RabbitMQ.
<code>rabbit_userid = guest</code>	(StrOpt) The RabbitMQ userid.
<code>rabbit_virtual_host = /</code>	(StrOpt) The RabbitMQ virtual host.

Table 2.2. Description of configuration options for kombu

Configuration option = Default value	Description
[DEFAULT]	
<code>kombu_reconnect_delay = 1.0</code>	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
<code>kombu_ssl_ca_certs =</code>	(StrOpt) SSL certification authority file (valid only if SSL enabled).
<code>kombu_ssl_certfile =</code>	(StrOpt) SSL cert file (valid only if SSL enabled).
<code>kombu_ssl_keyfile =</code>	(StrOpt) SSL key file (valid only if SSL enabled).
<code>kombu_ssl_version =</code>	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions.

Configure Qpid

Use these options to configure the Qpid messaging system for OpenStack Oslo RPC. Qpid is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `nova.conf` file.

```
rpc_backend=nova.openstack.common.rpc.impl_qpid
```

This critical option points the compute nodes to the Qpid broker (server). Set `qpid_hostname` to the host name where the broker runs in the `nova.conf` file.



Note

The `--qpid_hostname` option accepts a host name or IP address value.

```
qpid_hostname=hostname.example.com
```

If the Qpid broker listens on a port other than the AMQP default of 5672, you must set the `qpid_port` option to that value:

```
qpid_port=12345
```

If you configure the Qpid broker to require authentication, you must add a user name and password to the configuration:

```
qpid_username=username
qpid_password=password
```

By default, TCP is used as the transport. To enable SSL, set the `qpid_protocol` option:

```
qpid_protocol=ssl
```

This table lists additional options that you use to configure the Qpid messaging driver for OpenStack Oslo RPC. These options are used infrequently.

Table 2.3. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
<code>qpid_heartbeat = 60</code>	(IntOpt) Seconds between connection keepalive heartbeats.
<code>qpid_hostname = localhost</code>	(StrOpt) Qpid broker hostname.
<code>qpid_hosts = \$qpid_hostname:\$qpid_port</code>	(ListOpt) Qpid HA cluster host:port pairs.
<code>qpid_password =</code>	(StrOpt) Password for Qpid connection.
<code>qpid_port = 5672</code>	(IntOpt) Qpid broker port.
<code>qpid_protocol = tcp</code>	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
<code>qpid_sasl_mechanisms =</code>	(StrOpt) Space separated list of SASL mechanisms to use for auth.
<code>qpid_tcp_nodelay = True</code>	(BoolOpt) Whether to disable the Nagle algorithm.
<code>qpid_topology_version = 1</code>	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by <code>impl_qpid</code> . Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
<code>qpid_username =</code>	(StrOpt) Username for Qpid connection.

Configure ZeroMQ

Use these options to configure the ZeroMQ messaging system for OpenStack Oslo RPC. ZeroMQ is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `nova.conf` file.

Table 2.4. Description of configuration options for zeromq

Configuration option = Default value	Description
[DEFAULT]	
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1.
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets.
rpc_zmq_matchmaker = oslo.messaging_drivers.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver.
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port.
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Configure messaging

Use these options to configure the RabbitMQ and Qpid messaging drivers.

Table 2.5. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
control_exchange = openstack	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
rpc_backend = rabbit	(StrOpt) The messaging driver to use, defaults to rabbit. Other drivers include qpid and zmq.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC greenthread pool.
[cells]	
rpc_driver_queue_base = cells.intercell	(StrOpt) Base queue name to use when communicating between cells. Various topics by message type will be appended to this.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).
[upgrade_levels]	
baseapi = None	(StrOpt) Set a version cap for messages sent to the base api in any service

Configure the Compute API

The Compute API, run by the `nova-api` daemon, is the component of OpenStack Compute that receives and responds to user requests, whether they be direct API calls, or via the CLI tools or dashboard.

Configure Compute API password handling

The OpenStack Compute API enables users to specify an administrative password when they create or rebuild a server instance. If the user does not specify a password, a random password is generated and returned in the API response.

In practice, how the admin password is handled depends on the hypervisor in use and might require additional configuration of the instance. For example, you might have to install an agent to handle the password setting. If the hypervisor and instance configuration do not support setting a password at server create time, the password that is returned by the create API call is misleading because it was ignored.

To prevent this confusion, use the `enable_instance_password` configuration option to disable the return of the admin password for installations that do not support setting instance passwords.

Configure Compute API rate limiting

OpenStack Compute supports API rate limiting for the OpenStack API. The rate limiting allows an administrator to configure limits on the type and number of API calls that can be made in a specific time interval.

When API rate limits are exceeded, HTTP requests return an error with a status code of 413 `Request entity too large`, and includes an HTTP `Retry-After` header. The response body includes the error details and the delay before you should retry the request.

Rate limiting is not available for the EC2 API.

Define limits

To define limits, set these values:

- The **HTTP method** used in the API call, typically one of GET, PUT, POST, or DELETE.
- A **human readable URI** that is used as a friendly description of where the limit is applied.
- A **regular expression**. The limit is applied to all URIs that match the regular expression and HTTP method.
- A **limit value** that specifies the maximum count of units before the limit takes effect.
- An **interval** that specifies time frame to which the limit is applied. The interval can be SECOND, MINUTE, HOUR, or DAY.

Rate limits are applied in relative order to the HTTP method, going from least to most specific.

Default limits

Normally, you install OpenStack Compute with the following limits enabled:

Table 2.6. Default API rate limits

HTTP method	API URI	API regular expression	Limit
POST	any URI (*)	.*	120 per minute
POST	/servers	^/servers	120 per minute
PUT	any URI (*)	.*	120 per minute
GET	*changes-since*	.*changes-since.*	120 per minute
DELETE	any URI (*)	.*	120 per minute
GET	*/os-fping	^/os-fping	12 per minute

Configure and change limits

As part of the WSGI pipeline, the `etc/nova/api-paste.ini` file defines the actual limits.

To enable limits, include the `ratelimit` filter in the API pipeline specification. If the `ratelimit` filter is removed from the pipeline, limiting is disabled. You must also define the rate limit filter. The lines appear as follows:

```
[pipeline:openstack_compute_api_v2]
pipeline = faultwrap authtoken keystonecontext ratelimit osapi_compute_app_v2

[pipeline:openstack_volume_api_v1]
pipeline = faultwrap authtoken keystonecontext ratelimit osapi_volume_app_v1

[filter:ratelimit]
paste.filter_factory = nova.api.openstack.compute.
limits:RateLimitingMiddleware.factory
```

To modify the limits, add a `limits` specification to the `[filter:ratelimit]` section of the file. Specify the limits in this order:

1. HTTP method
2. friendly URI
3. regex
4. limit
5. interval

The following example shows the default rate-limiting values:

```
[filter:ratelimit]
paste.filter_factory = nova.api.openstack.compute.
limits:RateLimitingMiddleware.factory
limits = (POST, "*", .*, 120, MINUTE);(POST, "*/servers", ^/servers, 120,
MINUTE);(PUT, "*", .*, 120, MINUTE);(GET, "*changes-since*", .*changes-since.
*, 120, MINUTE);(DELETE, "*", .*, 120, MINUTE);(GET, "*/os-fping", ^/os-fping,
12, MINUTE)
```

Configuration reference

The Compute API configuration options are documented in [Table 2.11, “Description of configuration options for api” \[205\]](#).

Configure the EC2 API

You can set options in the `nova.conf` configuration file to control which network address and port the EC2 API listens on, the formatting of some API responses, and authentication related options.

To customize these options for OpenStack EC2 API, use the configuration option settings documented in [Table 2.25, “Description of configuration options for ec2” \[214\]](#).

Fibre Channel support in Compute

Fibre Channel support in OpenStack Compute is remote block storage attached to compute nodes for VMs.

In the Grizzly release, Fibre Channel supported only the KVM hypervisor.

Compute and Block Storage for Fibre Channel do not support automatic zoning. Fibre Channel arrays must be pre-zoned or directly attached to the KVM hosts.

KVM host requirements

You must install these packages on the KVM host:

- `sysfsutils` - Nova uses the `systool` application in this package.
- `sg3-utils` - Nova uses the `sg_scan` and `sginfo` applications.

Installing the `multipath-tools` package is optional.

Install required packages

Use these commands to install the system packages:

- For systems running openSUSE or SUSE Linux Enterprise Server:

```
# zypper install sysfsutils sg3_utils multipath-tools
```

- For systems running Red Hat:

```
# yum install sysfsutils sg3_utils multipath-tools
```

- For systems running Ubuntu:

```
# apt-get install sysfsutils sg3-utils multipath-tools
```

Hypervisors

OpenStack Compute supports many hypervisors, which might make it difficult for you to choose one. Most installations use only one hypervisor. However you can use [the section](#)

called “ComputeFilter” [183] and the section called “ImagePropertiesFilter” [185] to schedule to different hypervisors within the same installation. The following links help you choose a hypervisor. See <http://wiki.openstack.org/HypervisorSupportMatrix> for a detailed list of features and support across the hypervisors.

The following hypervisors are supported:

- **KVM** - Kernel-based Virtual Machine. The virtual disk formats that it supports is inherited from QEMU since it uses a modified QEMU program to launch the virtual machine. The supported formats include raw images, the qcow2, and VMware formats.
- **LXC** - Linux Containers (through libvirt), use to run Linux-based virtual machines.
- **QEMU** - Quick EMUlator, generally only used for development purposes.
- **UML** - User Mode Linux, generally only used for development purposes.
- **VMware vSphere** 4.1 update 1 and newer, runs VMware-based Linux and Windows images through a connection with a vCenter server or directly with an ESXi host.
- **Xen** - XenServer, Xen Cloud Platform (XCP), use to run Linux or Windows virtual machines. You must install the `nova-compute` service in a para-virtualized VM.
- **Hyper-V** - Server virtualization with Microsoft's Hyper-V, use to run Windows, Linux, and FreeBSD virtual machines. Runs `nova-compute` natively on the Windows virtualization platform.
- **Bare Metal** - Not a hypervisor in the traditional sense, this driver provisions physical hardware through pluggable sub-drivers (for example, PXE for image deployment, and IPMI for power management).

Hypervisor configuration basics

The node where the `nova-compute` service is installed and running is the machine that runs all the virtual machines, referred to as the compute node in this guide.

By default, the selected hypervisor is KVM. To change to another hypervisor, change the `virt_type` option in the `[libvirt]` section of `nova.conf` and restart the `nova-compute` service.

Here are the general `nova.conf` options that are used to configure the compute node's hypervisor: [Table 2.29, “Description of configuration options for hypervisor” \[215\]](#).

Specific options for particular hypervisors can be found in following sections.

KVM

KVM is configured as the default hypervisor for Compute.



Note

This document contains several sections about hypervisor selection. If you are reading this document linearly, you do not want to load the KVM module before you install `nova-compute`. The `nova-compute` service depends on

qemu-kvm, which installs `/lib/udev/rules.d/45-qemu-kvm.rules`, which sets the correct permissions on the `/dev/kvm` device node.

To enable KVM explicitly, add the following configuration options to the `/etc/nova/nova.conf` file:

```
compute_driver = libvirt.LibvirtDriver

[libvirt]
virt_type = kvm
```

The KVM hypervisor supports the following virtual machine image formats:

- Raw
- QEMU Copy-on-write (qcow2)
- QED Qemu Enhanced Disk
- VMWare virtual machine disk format (vmdk)

This section describes how to enable KVM on your system. For more information, see the following distribution-specific documentation:

- [Fedora: Getting started with virtualization](#) from the Fedora project wiki.
- [Ubuntu: KVM/Installation](#) from the Community Ubuntu documentation.
- [Debian: Virtualization with KVM](#) from the Debian handbook.
- [Red Hat Enterprise Linux: Installing virtualization packages on an existing Red Hat Enterprise Linux system](#) from the *Red Hat Enterprise Linux Virtualization Host Configuration and Guest Installation Guide*.
- [openSUSE: Installing KVM](#) from the openSUSE Virtualization with KVM manual.
- [SLES: Installing KVM](#) from the SUSE Linux Enterprise Server Virtualization with KVM manual.

Enable KVM

To perform these steps, you must be logged in as the `root` user.

1. To determine whether the `svm` or `vmx` CPU extensions are present, run this command:

```
# grep -E 'svm|vmx' /proc/cpuinfo
```

This command generates output if the CPU is hardware-virtualization capable. Even if output is shown, you might still need to enable virtualization in the system BIOS for full support.

If no output appears, consult your system documentation to ensure that your CPU and motherboard support hardware virtualization. Verify that any relevant hardware virtualization options are enabled in the system BIOS.

The BIOS for each manufacturer is different. If you must enable virtualization in the BIOS, look for an option containing the words `virtualization`, `VT`, `VMX`, or `SVM`.

- To list the loaded kernel modules and verify that the `kvm` modules are loaded, run this command:

```
# lsmod | grep kvm
```

If the output includes `kvm_intel` or `kvm_amd`, the `kvm` hardware virtualization modules are loaded and your kernel meets the module requirements for OpenStack Compute.

If the output does not show that the `kvm` module is loaded, run this command to load it:

```
# modprobe -a kvm
```

Run the command for your CPU. For Intel, run this command:

```
# modprobe -a kvm-intel
```

For AMD, run this command:

```
# modprobe -a kvm-amd
```

Because a KVM installation can change user group membership, you might need to log in again for changes to take effect.

If the kernel modules do not load automatically, use the procedures listed in these subsections.

If the checks indicate that required hardware virtualization support or kernel modules are disabled or unavailable, you must either enable this support on the system or find a system with this support.



Note

Some systems require that you enable VT support in the system BIOS. If you believe your processor supports hardware acceleration but the previous command did not produce output, reboot your machine, enter the system BIOS, and enable the VT option.

If KVM acceleration is not supported, configure Compute to use a different hypervisor, such as [QEMU](#) or [Xen](#).

These procedures help you load the kernel modules for Intel-based and AMD-based processors if they do not load automatically during KVM installation.

Intel-based processors

If your compute host is Intel-based, run these commands as root to load the kernel modules:

```
# modprobe kvm
# modprobe kvm-intel
```

Add these lines to the `/etc/modules` file so that these modules load on reboot:

```
kvm
kvm-intel
```

AMD-based processors

If your compute host is AMD-based, run these commands as root to load the kernel modules:

```
# modprobe kvm
# modprobe kvm-amd
```

Add these lines to `/etc/modules` file so that these modules load on reboot:

```
kvm
kvm-amd
```

Specify the CPU model of KVM guests

The Compute service enables you to control the guest CPU model that is exposed to KVM virtual machines. Use cases include:

- To maximize performance of virtual machines by exposing new host CPU features to the guest
- To ensure a consistent default CPU across all machines, removing reliance of variable QEMU defaults

In libvirt, the CPU is specified by providing a base CPU model name (which is a shorthand for a set of feature flags), a set of additional feature flags, and the topology (sockets/cores/threads). The libvirt KVM driver provides a number of standard CPU model names. These models are defined in the `/usr/share/libvirt/cpu_map.xml` file. Check this file to determine which models are supported by your local installation.

Two Compute configuration options in the `[libvirt]` group of `nova.conf` define which type of CPU model is exposed to the hypervisor when using KVM: `cpu_mode` and `cpu_model`.

The `cpu_mode` option can take one of the following values: `none`, `host-passthrough`, `host-model`, and `custom`.

Host model (default for KVM & QEMU)

If your `nova.conf` file contains `cpu_mode=host-model`, libvirt identifies the CPU model in `/usr/share/libvirt/cpu_map.xml` file that most closely matches the host, and requests additional CPU flags to complete the match. This configuration provides the maximum functionality and performance and maintains good reliability and compatibility if the guest is migrated to another host with slightly different host CPUs.

Host pass through

If your `nova.conf` file contains `cpu_mode=host-passthrough`, libvirt tells KVM to pass through the host CPU with no modifications. The difference to `host-model`, instead of just matching feature flags, every last detail of the host CPU is matched. This gives absolutely best performance, and can be important to some apps which check low level CPU details, but it comes at a cost with respect to migration: the guest can only be migrated to an exactly matching host CPU.

Custom

If your `nova.conf` file contains `cpu_mode=custom`, you can explicitly specify one of the supported named model using the `cpu_model` configuration option. For example, to configure the KVM guests to expose Nehalem CPUs, your `nova.conf` file should contain:

```
[libvirt]
cpu_mode = custom
cpu_model = Nehalem
```

None (default for all libvirt-driven hypervisors other than KVM & QEMU)

If your `nova.conf` file contains `cpu_mode=none`, libvirt does not specify a CPU model. Instead, the hypervisor chooses the default model.

Guest agent support

Use guest agents to enable optional access between compute nodes and guests through a socket, using the QMP protocol.

To enable this feature, you must set `hw_qemu_guest_agent=yes` as a metadata parameter on the image you wish to use to create guest-agent-capable instances from. You can explicitly disable the feature by setting `hw_qemu_guest_agent=no` in the image metadata.

KVM performance tweaks

The [VHostNet](#) kernel module improves network performance. To load the kernel module, run the following command as root:

```
# modprobe vhost_net
```

Troubleshoot KVM

Trying to launch a new virtual machine instance fails with the `ERRORstate`, and the following error appears in the `/var/log/nova/nova-compute.log` file:

```
libvirtError: internal error no supported architecture for os type 'hvm'
```

This message indicates that the KVM kernel modules were not loaded.

If you cannot start VMs after installation without rebooting, the permissions might not be correct. This can happen if you load the KVM module before you install `nova-compute`. To check whether the group is set to `kvm`, run:

```
# ls -l /dev/kvm
```

If it is not set to `kvm`, run:

```
# udevadm trigger
```

QEMU

From the perspective of the Compute service, the QEMU hypervisor is very similar to the KVM hypervisor. Both are controlled through libvirt, both support the same feature

set, and all virtual machine images that are compatible with KVM are also compatible with QEMU. The main difference is that QEMU does not support native virtualization. Consequently, QEMU has worse performance than KVM and is a poor choice for a production deployment.

The typical uses cases for QEMU are

- Running on older hardware that lacks virtualization support.
- Running the Compute service inside of a virtual machine for development or testing purposes, where the hypervisor does not support native virtualization for guests.

To enable QEMU, add these settings to `nova.conf`:

```
compute_driver = libvirt.LibvirtDriver

[libvirt]
virt_type = qemu
```

For some operations you may also have to install the **guestmount** utility:

On Ubuntu:

```
# apt-get install guestmount
```

On Red Hat Enterprise Linux, Fedora, or CentOS:

```
# yum install libguestfs-tools
```

On openSUSE:

```
# zypper install guestfs-tools
```

The QEMU hypervisor supports the following virtual machine image formats:

- Raw
- QEMU Copy-on-write (qcow2)
- VMware virtual machine disk format (vmdk)

Tips and fixes for QEMU on RHEL

If you are testing OpenStack in a virtual machine, you must configure Compute to use `qemu` without KVM and hardware virtualization. The second command relaxes SELinux rules to allow this mode of operation (https://bugzilla.redhat.com/show_bug.cgi?id=753589). The last two commands here work around a `libvirt` issue fixed in Red Hat Enterprise Linux 6.4. Nested virtualization will be the much slower TCG variety, and you should provide lots of memory to the top-level guest, because the OpenStack-created guests default to 2GM RAM with no overcommit.



Note

The second command, **setsebool**, may take a while.

```
# openstack-config --set /etc/nova/nova.conf libvirt virt_type qemu
# setsebool -P virt_use_execmem on
```

```
# ln -s /usr/libexec/qemu-kvm /usr/bin/qemu-system-x86_64
# service libvirtd restart
```

Xen, XenAPI, XenServer, and XCP



This section needs help

This section is low quality, and contains out of date information. The Documentation Team is currently looking for individuals with experience with the hypervisor to [Re-document Xen integration with OpenStack](#).

This section describes Xen, XenAPI, XenServer, and XCP, their differences, and how to use them with OpenStack. After you understand how the Xen and KVM architectures differ, you can determine when to use each architecture in your OpenStack cloud.

Xen terminology

Xen. A hypervisor that provides the fundamental isolation between virtual machines. Xen is open source (GPLv2) and is managed by Xen.org, an cross-industry organization.

Xen is a component of many different products and projects. The hypervisor itself is very similar across all these projects, but the way that it is managed can be different, which can cause confusion if you're not clear which tool stack you are using. Make sure you know what tool stack you want before you get started.

Xen Cloud Platform (XCP). An open source (GPLv2) tool stack for Xen. It is designed specifically as a platform for enterprise and cloud computing, and is well integrated with OpenStack. XCP is available both as a binary distribution, installed from an iso, and from Linux distributions, such as [xcp-xapi](#) in Ubuntu. The current versions of XCP available in Linux distributions do not yet include all the features available in the binary distribution of XCP.

Citrix XenServer. A commercial product. It is based on XCP, and exposes the same tool stack and management API. As an analogy, think of XenServer being based on XCP in the way that Red Hat Enterprise Linux is based on Fedora. XenServer has a free version (which is very similar to XCP) and paid-for versions with additional features enabled. Citrix provides support for XenServer, but as of July 2012, they do not provide any support for XCP. For a comparison between these products see the [XCP Feature Matrix](#).

Both XenServer and XCP include Xen, Linux, and the primary control daemon known as **xapi**.

The API shared between XCP and XenServer is called **XenAPI**. OpenStack usually refers to XenAPI, to indicate that the integration works equally well on XCP and XenServer. Sometimes, a careless person will refer to XenServer specifically, but you can be reasonably confident that anything that works on XenServer will also work on the latest version of XCP. Read the [XenAPI Object Model Overview](#) for definitions of XenAPI specific terms such as SR, VDI, VIF and PIF.

Privileged and unprivileged domains

A Xen host runs a number of virtual machines, VMs, or domains (the terms are synonymous on Xen). One of these is in charge of running the rest of the system, and is known as

"domain 0," or "dom0." It is the first domain to boot after Xen, and owns the storage and networking hardware, the device drivers, and the primary control software. Any other VM is unprivileged, and are known as a "domU" or "guest". All customer VMs are unprivileged of course, but you should note that on Xen the OpenStack control software (`nova-compute`) also runs in a domU. This gives a level of security isolation between the privileged system software and the OpenStack software (much of which is customer-facing). This architecture is described in more detail later.

There is an ongoing project to split domain 0 into multiple privileged domains known as **driver domains** and **stub domains**. This would give even better separation between critical components. This technology is what powers Citrix XenClient RT, and is likely to be added into XCP in the next few years. However, the current architecture just has three levels of separation: dom0, the OpenStack domU, and the completely unprivileged customer VMs.

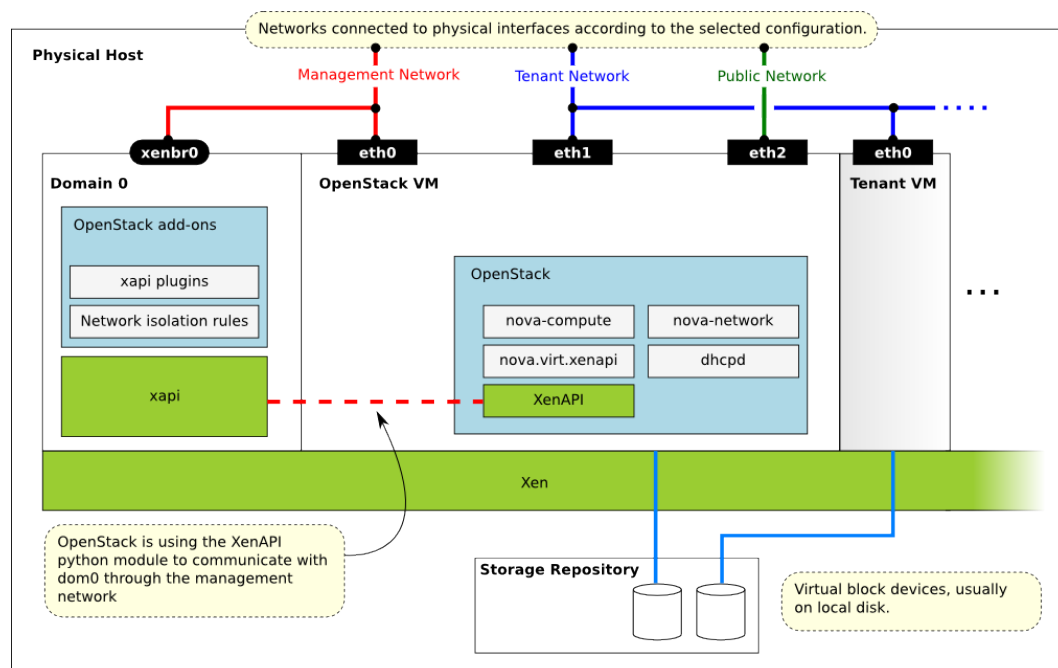
Paravirtualized versus hardware virtualized domains

A Xen virtual machine can be **paravirtualized (PV)** or **hardware virtualized (HVM)**. This refers to the interaction between Xen, domain 0, and the guest VM's kernel. PV guests are aware of the fact that they are virtualized and will co-operate with Xen and domain 0; this gives them better performance characteristics. HVM guests are not aware of their environment, and the hardware has to pretend that they are running on an unvirtualized machine. HVM guests do not need to modify the guest operating system, which is essential when running Windows.

In OpenStack, customer VMs may run in either PV or HVM mode. However, the OpenStack domU (that's the one running `nova-compute`) **must** be running in PV mode.

XenAPI Deployment Architecture

When you deploy OpenStack on XCP or XenServer, you get something similar to this:



Key things to note:

- The hypervisor: Xen
- Domain 0: runs xapi and some small pieces from OpenStack (some xapi plug-ins and network isolation rules). The majority of this is provided by XenServer or XCP (or yourself using Kronos).
- OpenStack VM: The `nova-compute` code runs in a paravirtualized virtual machine, running on the host under management. Each host runs a local instance of `nova-compute`. It will often also be running `nova-network` (depending on your network mode). In this case, `nova-network` is managing the addresses given to the tenant VMs through DHCP.
- Nova uses the XenAPI Python library to talk to xapi, and it uses the Management Network to reach from the domU to dom0 without leaving the host.

Some notes on the networking:

- The above diagram assumes FlatDHCP networking (the DevStack default).
- There are three main OpenStack Networks:
 - Management network - RabbitMQ, MySQL, etc. Please note that the VM images are downloaded by the XenAPI plug-ins, so make sure that the images can be downloaded through the management network. It usually means binding those services to the management interface.
 - Tenant network - controlled by `nova-network`. The parameters of this network depend on the networking model selected (Flat, Flat DHCP, VLAN).
 - Public network - floating IPs, public API endpoints.
- The networks shown here must be connected to the corresponding physical networks within the data center. In the simplest case, three individual physical network cards could be used. It is also possible to use VLANs to separate these networks. Please note, that the selected configuration must be in line with the networking model selected for the cloud. (In case of VLAN networking, the physical channels have to be able to forward the tagged traffic.)

XenAPI pools

The host-aggregates feature enables you to create pools of XenServer hosts to enable live migration when using shared storage. However, you cannot configure shared storage.

Further reading

Here are some of the resources available to learn more about Xen:

- Citrix XenServer official documentation: <http://docs.vmd.citrix.com/XenServer>.
- What is Xen? by Xen.org: <http://xen.org/files/Marketing/WhatisXen.pdf>.
- Xen Hypervisor project: <http://xen.org/products/xenhyp.html>.

- XCP project: <http://xen.org/products/cloudxen.html>.
- Further XenServer and OpenStack information: <http://wiki.openstack.org/XenServer>.

Install XenServer and XCP

Before you can run OpenStack with XCP or XenServer, you must install the software on an [appropriate server](#).



Note

Xen is a type 1 hypervisor: When your server starts, Xen is the first software that runs. Consequently, you must install XenServer or XCP before you install the operating system where you want to run OpenStack code. The OpenStack services then run in a virtual machine that you install on top of XenServer.

Before you can install your system, decide whether to install a free or paid edition of Citrix XenServer or Xen Cloud Platform from Xen.org. Download the software from these locations:

- <http://www.citrix.com/XenServer/download>
- <http://www.xen.org/download/xcp/index.html>

When you install many servers, you might find it easier to perform [PXE boot installations of XenServer or XCP](#). You can also package any post-installation changes that you want to make to your XenServer by [creating your own XenServer supplemental pack](#).

You can also install the xcp-xenapi package on Debian-based distributions to get XCP. However, this is not as mature or feature complete as above distributions. This modifies your boot loader to first boot Xen and boot your existing OS on top of Xen as Dom0. The xapi daemon runs in Dom0. Find more details at http://wiki.xen.org/wiki/Project_Kronos.



Important

Make sure you use the EXT type of storage repository (SR). Features that require access to VHD files (such as copy on write, snapshot and migration) do not work when you use the LVM SR. Storage repository (SR) is a XenAPI-specific term relating to the physical storage where virtual disks are stored.

On the XenServer/XCP installation screen, choose the **XenDesktop Optimized** option. If you use an answer file, make sure you use `srtype="ext"` in the `installation` tag of the answer file.

Post-installation steps

Complete these steps to install OpenStack in your XenServer system:

1. For resize and migrate functionality, complete the changes described in the *Configure resize* section in the [OpenStack Configuration Reference](#).
2. Install the VIF isolation rules to help prevent mac and IP address spoofing.

3. Install the XenAPI plug-ins. See the following section.
4. To support AMI type images, you must set up `/boot/guest symlink/directory` in Dom0. For detailed instructions, see next section.
5. To support resize/migration, set up an ssh trust relation between your XenServer hosts, and ensure `/images` is properly set up. See next section for more details.
6. Create a Paravirtualized virtual machine that can run the OpenStack compute code.
7. Install and configure the `nova-compute` in the above virtual machine.

For more information, see how DevStack performs the last three steps for developer deployments. For more information about DevStack, see *Getting Started With XenServer and Devstack* (<https://github.com/openstack-dev/devstack/blob/master/tools/xen/README.md>). Find more information about the first step, see *Multi Tenancy Networking Protections in XenServer* (<https://github.com/openstack/nova/blob/master/plugins/xenserver/doc/networking.rst>). For information about how to install the XenAPI plug-ins, see *XenAPI README* (<https://github.com/openstack/nova/blob/master/plugins/xenserver/xenapi/README>).

Install the XenAPI plug-ins

When you use Xen as the hypervisor for OpenStack Compute, you can install a Python script (or any executable) on the host side, and call that through the XenAPI. These scripts are called plug-ins. The XenAPI plug-ins live in the nova code repository. These plug-ins have to be copied to the Dom0 for the hypervisor, to the appropriate directory, where xapi can find them. There are several options for the installation. The important thing is to ensure that the version of the plug-ins are in line with the nova installation by only installing plug-ins from a matching nova repository.

Manually install the plug-in

1. Create temporary files/directories:

```
$ NOVA_ZIPBALL=$(mktemp)
$ NOVA_SOURCES=$(mktemp -d)
```

2. Get the source from github. The example assumes the master branch is used. Amend the URL to match the version being used:

```
$ wget -qO "$NOVA_ZIPBALL" https://github.com/openstack/nova/archive/
master.zip
$ unzip "$NOVA_ZIPBALL" -d "$NOVA_SOURCES"
```

(Alternatively) To use the official Ubuntu packages, use the following commands to get the nova code base:

```
$ ( cd $NOVA_SOURCES && apt-get source python-nova --download-only )
$ ( cd $NOVA_SOURCES && for ARCHIVE in *.tar.gz; do tar -xzf $ARCHIVE;
done )
```

3. Copy the plug-ins to the hypervisor:

```
$ PLUGINPATH=$(find $NOVA_SOURCES -path '*/xapi.d/plugins' -type d -print)
```

```
$ tar -czf - -C "$PLUGINPATH" ./ | ssh root@xenserver tar -xozf - -C /etc/xapi.d/plugins/
```

- Remove the temporary files/directories:

```
$ rm "$NOVA_ZIPBALL"
$ rm -rf "$NOVA_SOURCES"
```

Package a XenServer supplemental pack

Follow these steps to produce a supplemental pack from the nova sources, and package it as a XenServer supplemental pack.

- Create RPM packages. Given you have the nova sources. Use one of the methods in [the section called "Manually install the plug-in" \[159\]](#):

```
$ cd nova/plugins/xenserver/xenapi/contrib
$ ./build-rpm.sh
```

These commands leave an `.rpm` file in the `rpmbuild/RPMS/noarch/` directory.

- Pack the RPM packages to a Supplemental Pack, using the XenServer DDK (the following command should be issued on the XenServer DDK virtual appliance, after the produced rpm file has been copied over):

```
$ /usr/bin/build-supplemental-pack \  
> --output=output_directory \  
> --vendor-code=novaplugin \  
> --vendor-name=openstack \  
> --label=novaplugins \  
> --text="nova plugins" \  
> --version=0 \  
> full_path_to_rpmfile
```

This command produces an `.iso` file in the output directory specified. Copy that file to the hypervisor.

- Install the Supplemental Pack. Log in to the hypervisor, and issue:

```
# xe-install-supplemental-pack path_to_isofile
```

Prepare for AMI type images

To support AMI type images in your OpenStack installation, you must create a `/boot/guest` directory inside `Dom0`. The OpenStack VM extracts the kernel and ramdisk from the AKI and ARI images and puts them in this location.

OpenStack maintains the contents of this directory and its size should not increase during normal operation. However, in case of power failures or accidental shutdowns, some files might be left over. To prevent these files from filling the `Dom0` disk, set up this directory as a symlink that points to a subdirectory of the local SR.

Run these commands in `Dom0` to achieve this setup:

```
# LOCAL_SR=$(xe sr-list name-label="Local storage" --minimal)
# LOCALPATH="/var/run/sr-mount/$LOCAL_SR/os-guest-kernels"
# mkdir -p "$LOCALPATH"
# ln -s "$LOCALPATH" /boot/guest
```

Modify Dom0 for resize/migration support

To resize servers with XenServer and XCP, you must:

- Establish a root trust between all hypervisor nodes of your deployment:

To do so, generate an ssh key-pair with the `ssh-keygen` command. Ensure that each of your dom0's `authorized_keys` file (located in `/root/.ssh/authorized_keys`) contains the public key fingerprint (located in `/root/.ssh/id_rsa.pub`).

- Provide an `/images` mount point to the dom0 for your hypervisor:

Dom0 space is at a premium so creating a directory in dom0 is potentially dangerous and likely to fail especially when you resize large servers. The least you can do is to symlink `/images` to your local storage SR. The following instructions work for an English-based installation of XenServer (and XCP) and in the case of ext3-based SR (with which the resize functionality is known to work correctly).

```
# LOCAL_SR=$(xe sr-list name-label="Local storage" --minimal)
# IMG_DIR="/var/run/sr-mount/$LOCAL_SR/images"
# mkdir -p "$IMG_DIR"
# ln -s "$IMG_DIR" /images
```

Xen boot from ISO

XenServer, through the XenAPI integration with OpenStack, provides a feature to boot instances from an ISO file. To activate the Boot From ISO feature, you must configure the SR elements on XenServer host, as follows:

1. Create an ISO-typed SR, such as an NFS ISO library, for instance. For this, using XenCenter is a simple method. You must export an NFS volume from a remote NFS server. Make sure it is exported in read-write mode.
2. On the compute host, find and record the uuid of this ISO SR:

```
# xe host-list
```

3. Locate the uuid of the NFS ISO library:

```
# xe sr-list content-type=iso
```

4. Set the uuid and configuration. Even if an NFS mount point is not local, you must specify `local-storage-iso`.

```
# xe sr-param-set uuid=[iso sr uuid] other-config:i18n-key=local-storage-iso
```

5. Make sure the host-uuid from `xe pbd-list` equals the uuid of the host you found previously:

```
# xe sr-uuid=[iso sr uuid]
```

6. You can now add images through the OpenStack Image Service with `disk-format=iso`, and boot them in OpenStack Compute:

```
$ glance image-create --name=fedora_iso --disk-format=iso --container-format=bare < Fedora-16-x86_64-netinst.iso
```

Xen configuration reference

The following section discusses some commonly changed options in XenServer. The table below provides a complete reference of all configuration options available for configuring Xen with OpenStack.

The recommended way to use Xen with OpenStack is through the XenAPI driver. To enable the XenAPI driver, add the following configuration options `/etc/nova/nova.conf` and restart the `nova-compute` service:

```
compute_driver = xenapi.XenAPIDriver
xenapi_connection_url = http://your_xenapi_management_ip_address
xenapi_connection_username = root
xenapi_connection_password = your_password
```

These connection details are used by the OpenStack Compute service to contact your hypervisor and are the same details you use to connect XenCenter, the XenServer management console, to your XenServer or XCP box.



Note

The `xenapi_connection_url` is generally the management network IP address of the XenServer. Though it is possible to use the internal network IP Address (169.250.0.1) to contact XenAPI, this does not allow live migration between hosts, and other functionalities like host aggregates do not work.

It is possible to manage Xen using libvirt, though this is not well-tested or supported. To experiment using Xen through libvirt add the following configuration options `/etc/nova/nova.conf`:

```
compute_driver = libvirt.LibvirtDriver

[libvirt]
virt_type = xen
```

Agent

If you don't have the guest agent on your VMs, it takes a long time for nova to decide the VM has successfully started. Generally a large timeout is required for Windows instances, but you may want to tweak `agent_version_timeout`

Firewall

If using nova-network, IPTables is supported:

```
firewall_driver = nova.virt.firewall.IptablesFirewallDriver
```

Alternately, doing the isolation in Dom0:

```
firewall_driver = nova.virt.xenapi.firewall.Dom0IptablesFirewallDriver
```

VNC proxy address

Assuming you are talking to XenAPI through the host local management network, and XenServer is on the address: 169.254.0.1, you can use the following:
`vncserver_proxycient_address=169.254.0.1`

Storage

You can specify which Storage Repository to use with nova by looking at the following flag. The default is to use the local-storage setup by the default installer:

```
sr_matching_filter = "other-config:i18n-key=local-storage"
```

Another good alternative is to use the "default" storage (for example if you have attached NFS or any other shared storage):

```
sr_matching_filter = "default-sr:true"
```



Note

To use a XenServer pool, you must create the pool by using the Host Aggregates feature.

Xen configuration reference

To customize the Xen driver, use the configuration option settings documented in [Table 2.59, "Description of configuration options for xen" \[233\]](#).

LXC (Linux containers)

LXC (also known as Linux containers) is a virtualization technology that works at the operating system level. This is different from hardware virtualization, the approach used by other hypervisors such as KVM, Xen, and VMware. LXC (as currently implemented using libvirt in the Compute service) is not a secure virtualization technology for multi-tenant environments (specifically, containers may affect resource quotas for other containers hosted on the same machine). Additional containment technologies, such as AppArmor, may be used to provide better isolation between containers, although this is not the case by default. For all these reasons, the choice of this virtualization technology is not recommended in production.

If your compute hosts do not have hardware support for virtualization, LXC will likely provide better performance than QEMU. In addition, if your guests must access specialized hardware, such as GPUs, this might be easier to achieve with LXC than other hypervisors.



Note

Some OpenStack Compute features might be missing when running with LXC as the hypervisor. See the [hypervisor support matrix](#) for details.

To enable LXC, ensure the following options are set in `/etc/nova/nova.conf` on all hosts running the `nova-compute` service.

```
compute_driver = libvirt.LibvirtDriver

[libvirt]
virt_type = lxc
```

On Ubuntu 12.04, enable LXC support in OpenStack by installing the `nova-compute-lxc` package.

VMware vSphere

Introduction

OpenStack Compute supports the VMware vSphere product family and enables access to advanced features such as vMotion, High Availability, and Dynamic Resource Scheduling (DRS). This section describes how to configure VMware-based virtual machine images for launch. vSphere versions 4.1 and newer are supported.

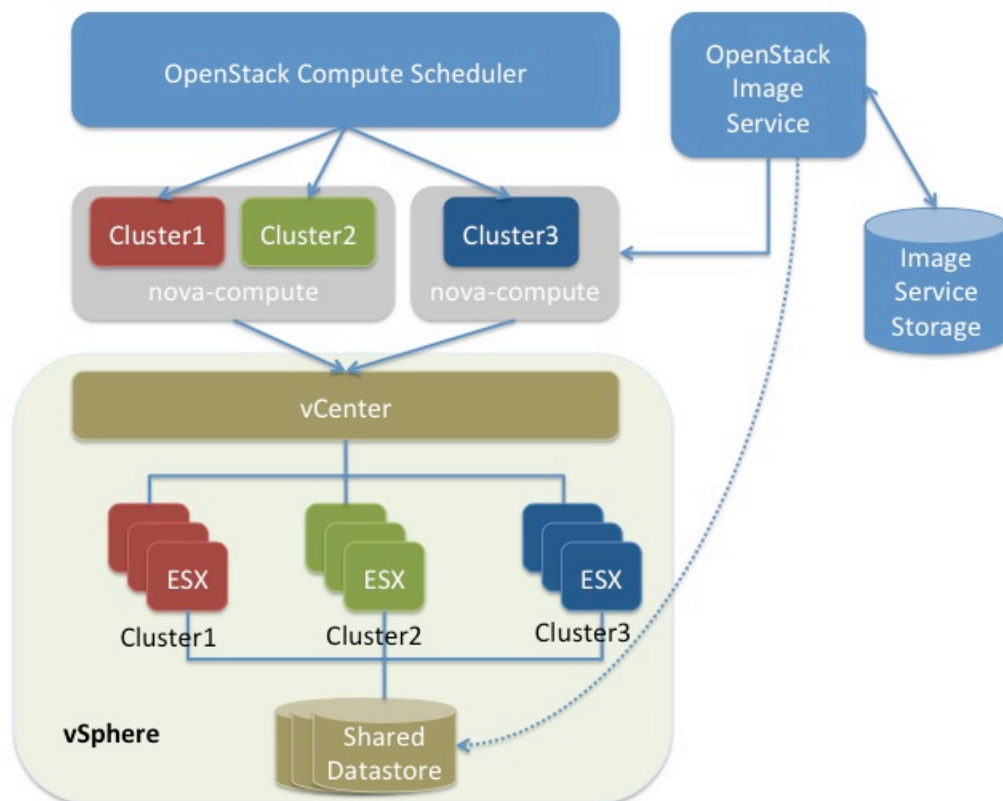
The VMware vCenter driver enables the `nova-compute` service to communicate with a VMware vCenter server that manages one or more ESX host clusters. The driver aggregates the ESX hosts in each cluster to present one large hypervisor entity for each cluster to the Compute scheduler. Because individual ESX hosts are not exposed to the scheduler, Compute schedules to the granularity of clusters and vCenter uses DRS to select the actual ESX host within the cluster. When a virtual machine makes its way into a vCenter cluster, it can use all vSphere features.

The following sections describe how to configure the VMware vCenter driver.

High-level architecture

The following diagram shows a high-level view of the VMware driver architecture:

Figure 2.1. VMware driver architecture



As the figure shows, the OpenStack Compute Scheduler sees three hypervisors that each correspond to a cluster in vCenter. `nova-compute` contains the VMware driver. You can run with multiple `nova-compute` services. While Compute schedules at the granularity of a cluster, the VMware driver inside `nova-compute` interacts with the vCenter APIs to select an appropriate ESX host within the cluster. Internally, vCenter uses DRS for placement.

The VMware vCenter driver also interacts with the OpenStack Image Service to copy VMDK images from the Image Service back end store. The dotted line in the figure represents VMDK images being copied from the OpenStack Image Service to the vSphere data store. VMDK images are cached in the data store so the copy operation is only required the first time that the VMDK image is used.

After OpenStack boots a VM into a vSphere cluster, the VM becomes visible in vCenter and can access vSphere advanced features. At the same time, the VM is visible in the OpenStack dashboard and you can manage it as you would any other OpenStack VM. You can perform advanced vSphere operations in vCenter while you configure OpenStack resources such as VMs through the OpenStack dashboard.

The figure does not show how networking fits into the architecture. Both `nova-network` and the OpenStack Networking Service are supported. For details, see [the section called "Networking with VMware vSphere" \[171\]](#).

Configuration overview

To get started with the VMware vCenter driver, complete the following high-level steps:

1. Configure vCenter correctly. See [the section called "Prerequisites and limitations" \[165\]](#).
2. Configure `nova.conf` for the VMware vCenter driver. See [the section called "VMware vCenter driver" \[166\]](#).
3. Load desired VMDK images into the OpenStack Image Service. See [the section called "Images with VMware vSphere" \[167\]](#).
4. Configure networking with either `nova-network` or the OpenStack Networking Service. See [the section called "Networking with VMware vSphere" \[171\]](#).

Prerequisites and limitations

Use the following list to prepare a vSphere environment that runs with the VMware vCenter driver:

1. **Copying VMDK files (vSphere 5.1 only).** In vSphere 5.1, copying large image files (for example, 12 GB and greater) from Glance can take a long time. To improve performance, VMware recommends that you upgrade to VMware vCenter Server 5.1 Update 1 or later. For more information, see the [Release Notes](#).
2. **DRS.** For any cluster that contains multiple ESX hosts, enable DRS and enable fully automated placement.
3. **Shared storage.** Only shared storage is supported and data stores must be shared among all hosts in a cluster. It is recommended to remove data stores not intended for OpenStack from clusters being configured for OpenStack.

4. **Clusters and data stores.** Do not use OpenStack clusters and data stores for other purposes. If you do, OpenStack displays incorrect usage information.
5. **Networking.** The networking configuration depends on the desired networking model. See [the section called "Networking with VMware vSphere" \[171\]](#).
6. **Security groups.** If you use the VMware driver with OpenStack Networking and the NSX plug-in, security groups are supported. If you use `nova-network`, security groups are not supported.



Note

The NSX plug-in is the only plug-in that is validated for vSphere.

7. **VNC.** The port range 5900 - 6105 (inclusive) is automatically enabled for VNC connections on every ESX host in all clusters under OpenStack control. For more information about using a VNC client to connect to virtual machine, see http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1246.



Note

In addition to the default VNC port numbers (5900 to 6000) specified in the above document, the following ports are also used: 6101, 6102, and 6105.

You must modify the ESXi firewall configuration to allow the VNC ports. Additionally, for the firewall modifications to persist after a reboot, you must create a custom vSphere Installation Bundle (VIB) which is then installed onto the running ESXi host or added to a custom image profile used to install ESXi hosts. For details about how to create a VIB for persisting the firewall configuration modifications, see http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2007381.

8. **Ephemeral Disks.** Ephemeral disks are not supported. A future major release will address this limitation.
9. Injection of SSH keys into compute instances hosted by vCenter is not currently supported.
10. To use multiple vCenter installations with OpenStack, each vCenter must be assigned to a separate availability zone. This is required as the OpenStack Block Storage VMDK driver does not currently work across multiple vCenter installations.

VMware vCenter driver

Use the VMware vCenter driver (`VMwareVCDriver`) to connect OpenStack Compute with vCenter. This recommended configuration enables access through vCenter to advanced vSphere features like vMotion, High Availability, and Dynamic Resource Scheduling (DRS).

VMwareVCDriver configuration options

When you use the `VMwareVCDriver` (vCenter versions 5.1 and later) with OpenStack Compute, add the following VMware-specific configuration options to the `nova.conf` file:

```
[DEFAULT]
compute_driver=vmwareapi.VMwareVCDriver

[vmware]
host_ip=<vCenter host IP>
host_username=<vCenter username>
host_password=<vCenter password>
cluster_name=<vCenter cluster name>
datastore_regex=<optional datastore regex>
```



Note

- vSphere vCenter versions 5.0 and earlier: You must specify the location of the WSDL files by adding the `wsdl_location=http://127.0.0.1:8080/vmware/SDK/wsdl/vim25/vimService.wsdl` setting to the above configuration. For more information, see [vSphere 5.0 and earlier additional set up](#).
- Clusters: The vCenter driver can support multiple clusters. To use more than one cluster, simply add multiple `cluster_name` lines in `nova.conf` with the appropriate cluster name. Clusters and data stores used by the vCenter driver should not contain any VMs other than those created by the driver.
- Data stores: The `datastore_regex` setting specifies the data stores to use with Compute. For example, `datastore_regex="nas.*"` selects all the data stores that have a name starting with "nas". If this line is omitted, Compute uses the first data store returned by the vSphere API. It is recommended not to use this field and instead remove data stores that are not intended for OpenStack.
- Reserved host memory: The `reserved_host_memory_mb` option value is 512 MB by default. However, VMware recommends that you set this option to 0 MB because the vCenter driver reports the effective memory available to the virtual machines.

A `nova-compute` service can control one or more clusters containing multiple ESX hosts, making `nova-compute` a critical service from a high availability perspective. Because the host that runs `nova-compute` can fail while the vCenter and ESX still run, you must protect the `nova-compute` service against host failures.



Note

Many `nova.conf` options are relevant to libvirt but do not apply to this driver.

You must complete additional configuration for environments that use vSphere 5.0 and earlier. See [the section called “vSphere 5.0 and earlier additional set up” \[172\]](#).

Images with VMware vSphere

The vCenter driver supports images in the VMDK format. Disks in this format can be obtained from VMware Fusion or from an ESX environment. It is also possible to convert other formats, such as `qcow2`, to the VMDK format using the `qemu-img` utility. After a VMDK disk is available, load it into the OpenStack Image Service. Then, you can use it


```
-O vmdk precise-server-cloudimg-amd64-disk1.vmdk
```

VMDK disks converted through `qemu-img` are *always* monolithic sparse VMDK disks with an IDE adapter type. Using the previous example of the Precise Ubuntu image after the `qemu-img` conversion, the command to upload the VMDK disk should be something like:

```
$ glance image-create --name precise-cloud --is-public=True \  
--container-format=bare --disk-format=vmdk \  
--property vmware_disktype="sparse" \  
--property vmware_adaptertype="ide" < \  
precise-server-cloudimg-amd64-disk1.vmdk
```

Note that the `vmware_disktype` is set to *sparse* and the `vmware_adaptertype` is set to *ide* in the previous command.

If the image did not come from the `qemu-img` utility, the `vmware_disktype` and `vmware_adaptertype` might be different. To determine the image adapter type from an image file, use the following command and look for the `ddb.adapterType=` line:

```
$ head -20 <vmdk file name>
```

Assuming a preallocated disk type and an iSCSI lsiLogic adapter type, the following command uploads the VMDK disk:

```
$ glance image-create --name="ubuntu-thick-scsi" --disk_format=vmdk \  
--container_format=bare --is_public=true \  
--property vmware_adaptertype="lsiLogic" \  
--property vmware_disktype="preallocated" \  
--property vmware_ostype="ubuntu64Guest" < ubuntuLTS-flat.vmdk
```

Currently, OS boot VMDK disks with an IDE adapter type cannot be attached to a virtual SCSI controller and likewise disks with one of the SCSI adapter types (such as, `busLogic`, `lsiLogic`) cannot be attached to the IDE controller. Therefore, as the previous examples show, it is important to set the `vmware_adaptertype` property correctly. The default adapter type is `lsiLogic`, which is SCSI, so you can omit the `vmware_adaptertype` property if you are certain that the image adapter type is `lsiLogic`.

Tag VMware images

In a mixed hypervisor environment, OpenStack Compute uses the `hypervisor_type` tag to match images to the correct hypervisor type. For VMware images, set the `hypervisor_type` to `vmware`. Other valid hypervisor types include: `xen`, `qemu`, `kvm`, `lxc`, `uml`, and `hyperv`.

```
$ glance image-create --name="ubuntu-thick-scsi" --disk_format=vmdk \  
--container_format=bare --is_public=true \  
--property vmware_adaptertype="lsiLogic" \  
--property vmware_disktype="preallocated" \  
--property hypervisor_type="vmware" \  
--property vmware_ostype="ubuntu64Guest" < ubuntuLTS-flat.vmdk
```

Optimize images

Monolithic Sparse disks are considerably faster to download but have the overhead of an additional conversion step. When imported into ESX, sparse disks get converted to VMFS flat thin provisioned disks. The download and conversion steps only affect the first

launched instance that uses the sparse disk image. The converted disk image is cached, so subsequent instances that use this disk image can simply use the cached version.

To avoid the conversion step (at the cost of longer download times) consider converting sparse disks to thin provisioned or preallocated disks before loading them into the OpenStack Image Service. Below are some tools that can be used to pre-convert sparse disks.

1. Using vSphere CLI (or sometimes called the remote CLI or rCLI) tools

Assuming that the sparse disk is made available on a data store accessible by an ESX host, the following command converts it to preallocated format:

```
vmkfstools --server=ip_of_some_ESX_host -i /vmfs/volumes/datastore1/sparse.vmdk /vmfs/volumes/datastore1/converted.vmdk
```

(Note that the vifs tool from the same CLI package can be used to upload the disk to be converted. The vifs tool can also be used to download the converted disk if necessary.)

2. Using vmkfstools directly on the ESX host

If the SSH service is enabled on an ESX host, the sparse disk can be uploaded to the ESX data store via scp and the vmkfstools local to the ESX host can use used to perform the conversion: (After logging in to the host via ssh)

```
vmkfstools -i /vmfs/volumes/datastore1/sparse.vmdk /vmfs/volumes/datastore1/converted.vmdk
```

3. vmware-vdiskmanager

vmware-vdiskmanager is a utility that comes bundled with VMware Fusion and VMware Workstation. Below is an example of converting a sparse disk to preallocated format:

```
'/Applications/VMware Fusion.app/Contents/Library/vmware-vdiskmanager' -r sparse.vmdk -t 4 converted.vmdk
```

In all of the above cases, the converted vmdk is actually a pair of files: the descriptor file *converted.vmdk* and the actual virtual disk data file *converted-flat.vmdk*. The file to be uploaded to the OpenStack Image Service is *converted-flat.vmdk*.

Image handling

The ESX hypervisor requires a copy of the VMDK file in order to boot up a virtual machine. As a result, the vCenter OpenStack Compute driver must download the VMDK via HTTP from the OpenStack Image Service to a data store that is visible to the hypervisor. To optimize this process, the first time a VMDK file is used, it gets cached in the data store. Subsequent virtual machines that need the VMDK use the cached version and don't have to copy the file again from the OpenStack Image Service.

Even with a cached VMDK, there is still a copy operation from the cache location to the hypervisor file directory in the shared data store. To avoid this copy, boot the image in linked_clone mode. To learn how to enable this mode, see [the section called "Configuration reference" \[174\]](#). Note also that it is possible to override the linked_clone mode on a per-image basis by using the `vmware_linked_clone` property in the OpenStack Image Service.

You can configure the `nova.conf` file to automatically purge unused images after a specified period of time. The relevant settings in the `DEFAULT` section are:

- `remove_unused_base_images` - Set this parameter to `True` to specify that unused images should be removed after the duration specified in the `remove_unused_original_minimum_age_seconds` parameter. The default is `True`.
- `remove_unused_original_minimum_age_seconds` - Specifies the duration in seconds after which an unused image is purged from the cache. The default is `86400` (24 hours).

Networking with VMware vSphere

The VMware driver supports networking with the `nova-network` service or the OpenStack Networking Service. Depending on your installation, complete these configuration steps before you provision VMs:

- **The `nova-network` service with the FlatManager or FlatDHCPManager.** Create a port group with the same name as the `flat_network_bridge` value in the `nova.conf` file. The default value is `br100`. If you specify another value, the new value must be a valid linux bridge identifier that adheres to linux bridge naming conventions.

All VM NICs are attached to this port group.

Ensure that the flat interface of the node that runs the `nova-network` service has a path to this network.



Note

When configuring the port binding for this port group in vCenter, specify `ephemeral` for the port binding type. For more information, see [Choosing a port binding type in ESX/ESXi](#) in the VMware Knowledge Base.

- **The `nova-network` service with the VlanManager.** Set the `vlan_interface` configuration option to match the ESX host interface that handles VLAN-tagged VM traffic.

OpenStack Compute automatically creates the corresponding port groups.

- If you are using the OpenStack Networking Service: Before provisioning VMs, create a port group with the same name as the `vmware.integration_bridge` value in `nova.conf` (default is `br-int`). All VM NICs are attached to this port group for management by the OpenStack Networking plug-in.

Volumes with VMware vSphere

The VMware driver supports attaching volumes from the OpenStack Block Storage service. The VMware VMDK driver for OpenStack Block Storage is recommended and should be used for managing volumes based on vSphere data stores. More information about the VMware VMDK driver can be found at: [VMware VMDK Driver](#). Also an `iscsi` volume driver provides limited support and can be used only for attachments.

vSphere 5.0 and earlier additional set up

Users of vSphere 5.0 or earlier must host their WSDL files locally. These steps are applicable for vCenter 5.0 or ESXi 5.0 and you can either mirror the WSDL from the vCenter or ESXi server that you intend to use or you can download the SDK directly from VMware. These workaround steps fix a [known issue](#) with the WSDL that was resolved in later versions.

When setting the VMwareVCDriver configuration options, you must include the `wSDL_location` option. For more information, see [VMwareVCDriver configuration options](#) above.

Procedure 2.1. Mirror WSDL from vCenter (or ESXi)

1. Set the `VMWAREAPI_IP` shell variable to the IP address for your vCenter or ESXi host from where you plan to mirror files. For example:

```
$ export VMWAREAPI_IP=<your_vsphere_host_ip>
```

2. Create a local file system directory to hold the WSDL files:

```
$ mkdir -p /opt/stack/vmware/wSDL/5.0
```

3. Change into the new directory.

```
$ cd /opt/stack/vmware/wSDL/5.0
```

4. Use your OS-specific tools to install a command-line tool that can download files like **wget**.

5. Download the files to the local file cache:

```
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/vimService.wsdl
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/vim.wsdl
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/core-types.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/query-messages.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/query-types.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/vim-messages.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/vim-types.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/reflect-
messages.xsd
wget --no-check-certificate https://$VMWAREAPI_IP/sdk/reflect-types.xsd
```

Because the `reflect-types.xsd` and `reflect-messages.xsd` files do not fetch properly, you must stub out these files. Use the following XML listing to replace the missing file content. The XML parser underneath Python can be very particular and if you put a space in the wrong place, it can break the parser. Copy the following contents and formatting carefully.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="urn:reflect"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
</schema>
```

- Now that the files are locally present, tell the driver to look for the SOAP service WSDLs in the local file system and not on the remote vSphere server. Add the following setting to the `nova.conf` file for your `nova-compute` node:

```
[vmware]
wsdl_location=file:///opt/stack/vmware/wsdl/5.0/vimService.wsdl
```

Alternatively, download the version appropriate SDK from <http://www.vmware.com/support/developer/vc-sdk/> and copy it to the `/opt/stack/vmware` file. Make sure that the WSDL is available, in for example `/opt/stack/vmware/SDK/wsdl/vim25/vimService.wsdl`. You must point `nova.conf` to fetch this WSDL file from the local file system by using a URL.

When using the `VMwareVCDriver` (vCenter) with OpenStack Compute with vSphere version 5.0 or earlier, `nova.conf` must include the following extra config option:

```
[vmware]
wsdl_location=file:///opt/stack/vmware/SDK/wsdl/vim25/vimService.wsdl
```

VMware ESX driver

This section covers details of using the `VMwareESXDriver`. The ESX Driver has not been extensively tested and is not recommended. To configure the VMware vCenter driver instead, see [the section called "VMware vCenter driver" \[166\]](#).



Warning

The VMWare ESX driver has been deprecated in the Icehouse release and will be removed with the Juno release.

VMwareESXDriver configuration options

When you use the `VMwareESXDriver` (no vCenter) with OpenStack Compute, add the following VMware-specific configuration options to the `nova.conf` file:

```
[DEFAULT]
compute_driver=vmwareapi.VMwareESXDriver

[vmware]
host_ip=<ESXi host IP>
host_username=<ESXi host username>
host_password=<ESXi host password>
wsdl_location=http://127.0.0.1:8080/vmware/SDK/wsdl/vim25/vimService.wsdl
```

Remember that you will have one `nova-compute` service for each ESXi host. It is recommended that this host run as a VM on the same ESXi host that it manages.



Note

Many `nova.conf` options are relevant to libvirt but do not apply to this driver.

Requirements and limitations

The `ESXDriver` cannot use many of the vSphere platform advanced capabilities, namely vMotion, high availability, and DRS.

Configuration reference

To customize the VMware driver, use the configuration option settings documented in [Table 2.54, "Description of configuration options for vmware" \[230\]](#).

Hyper-V virtualization platform

It is possible to use Hyper-V as a compute node within an OpenStack Deployment. The `nova-compute` service runs as "openstack-compute," a 32-bit service directly upon the Windows platform with the Hyper-V role enabled. The necessary Python components as well as the `nova-compute` service are installed directly onto the Windows platform. Windows Clustering Services are not needed for functionality within the OpenStack infrastructure. The use of the Windows Server 2012 platform is recommend for the best experience and is the platform for active development. The following Windows platforms have been tested as compute nodes:

- **Windows Server 2008r2**

Both Server and Server Core with the Hyper-V role enabled (Shared Nothing Live migration is not supported using 2008r2)

- **Windows Server 2012**

Server and Core (with the Hyper-V role enabled), and Hyper-V Server

Hyper-V configuration

The following sections discuss how to prepare the Windows Hyper-V node for operation as an OpenStack compute node. Unless stated otherwise, any configuration information should work for both the Windows 2008r2 and 2012 platforms.

Local Storage Considerations

The Hyper-V compute node needs to have ample storage for storing the virtual machine images running on the compute nodes. You may use a single volume for all, or partition it into an OS volume and VM volume. It is up to the individual deploying to decide.

Configure NTP

Network time services must be configured to ensure proper operation of the Hyper-V compute node. To set network time on your Hyper-V host you must run the following commands:

```
C:\>net stop w32time
C:\>w32tm /config /manualpeerlist:pool.ntp.org,0x8 /syncfromflags:MANUAL
C:\>net start w32time
```

Configure Hyper-V virtual switching

Information regarding the Hyper-V virtual Switch can be located here: <http://technet.microsoft.com/en-us/library/hh831823.aspx>

To quickly enable an interface to be used as a Virtual Interface the following PowerShell may be used:

```
PS C:\>$if = Get-NetIPAddress -IPAddress 192* | Get-NetIPInterface
PS C:\>New-VMSwitch -NetAdapterName $if.ifAlias -Name yourbridgename -
AllowManagementOS $false
```

Enable iSCSI initiator service

To prepare the Hyper-V node to be able to attach to volumes provided by cinder you must first make sure the Windows iSCSI initiator service is running and started automatically.

```
C:\>sc start MSiSCSI
C:\>sc config MSiSCSI start="auto"
```

Configure shared nothing live migration

Detailed information on the configuration of live migration can be found here: <http://technet.microsoft.com/en-us/library/jj134199.aspx>

The following outlines the steps of shared nothing live migration.

1. The target hosts ensures that live migration is enabled and properly configured in Hyper-V.
2. The target hosts checks if the image to be migrated requires a base VHD and pulls it from the Image Service if not already available on the target host.
3. The source hosts ensures that live migration is enabled and properly configured in Hyper-V.
4. The source hosts initiates a Hyper-V live migration.
5. The source hosts communicates to the manager the outcome of the operation.

The following two configuration options/flags are needed in order to support Hyper-V live migration and must be added to your `nova.conf` on the Hyper-V compute node:

- `instances_shared_storage=False`

This needed to support "shared nothing" Hyper-V live migrations. It is used in `nova/compute/manager.py`

- `limit_cpu_features=True`

This flag is needed to support live migration to hosts with different CPU features. This flag is checked during instance creation in order to limit the CPU features used by the VM.

- `instances_path=DRIVELETTER:\PATH\TO\YOUR\INSTANCES`

Additional Requirements:

- Hyper-V 2012 RC or Windows Server 2012 RC with Hyper-V role enabled

- A Windows domain controller with the Hyper-V compute nodes as domain members
- The `instances_path` command-line option/flag needs to be the same on all hosts.
- The `openstack-compute` service deployed with the setup must run with domain credentials. You can set the service credentials with:

```
C:\>sc config openstack-compute obj="DOMAIN\username" password="password"
```

How to setup live migration on Hyper-V

To enable 'shared nothing live' migration, run the 3 PowerShell instructions below on each Hyper-V host:

```
PS C:\>Enable-VMMigration
PS C:\>Set-VMMigrationNetwork IP_ADDRESS
PS C:\>Set-VMHost -VirtualMachineMigrationAuthenticationTypeKerberos
```



Note

Please replace the `IP_ADDRESS` with the address of the interface which will provide the virtual switching for nova-network.

Additional Reading

Here's an article that clarifies the various live migration options in Hyper-V:

<http://ariessysadmin.blogspot.ro/2012/04/hyper-v-live-migration-of-windows.html>

Python Requirements

Python

Python 2.7.3 must be installed prior to installing the OpenStack Compute Driver on the Hyper-V server. Download and then install the MSI for windows here:

- <http://www.python.org/ftp/python/2.7.3/python-2.7.3.msi>
- Install the MSI accepting the default options.
- The installation will put python in `C:/python27`.

Setuptools

You will require pip to install the necessary python module dependencies. The installer will install under the `C:\python27` directory structure. Setuptools for Python 2.7 for Windows can be download from here:

<http://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11.win32-py2.7.exe>

Python Dependencies

You must download and manually install the following packages on the Compute node:

- **MySQL-python**

<http://codegood.com/download/10/>

- **pywin32**

Download and run the installer from the following location

<http://sourceforge.net/projects/pywin32/files/pywin32/Build%20217/pywin32-217.win32-py2.7.exe>

- **greenlet**

Select the link below:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

You must scroll to the greenlet section for the following file: greenlet-0.4.0.win32-py2.7.#exe

Click on the file, to initiate the download. Once the download is complete, run the installer.

You must install the following Python packages through **easy_install** or **pip**. Run the following replacing PACKAGENAME with the following packages:

```
C:\>c:\Python27\Scripts\pip.exe install PACKAGE_NAME
```

- amqp
- anyjson
- distribute
- eventlet
- httplib2
- iso8601
- jsonschema
- kombu
- netaddr
- paste
- paste-deploy
- prettytable
- python-cinderclient
- python-glanceclient
- python-keystoneclient

- repoze.lru
- routes
- sqlalchemy
- simplejson
- warlock
- webob
- wmi

Install Nova-compute

Using git on Windows to retrieve source

Git be used to download the necessary source code. The installer to run Git on Windows can be downloaded here:

<http://code.google.com/p/msysgit/downloads/list?q=full+installer+official+git>

Download the latest installer. Once the download is complete double click the installer and follow the prompts in the installation wizard. The default should be acceptable for the needs of the document.

Once installed you may run the following to clone the Nova code.

```
C:\>git.exe clone https://github.com/openstack/nova.git
```

Configure Nova.conf

The `nova.conf` file must be placed in `C:\etc\nova` for running OpenStack on Hyper-V. Below is a sample `nova.conf` for Windows:

```
[DEFAULT]
verbose=true
force_raw_images=false
auth_strategy=keystone
fake_network=true
vswitch_name=openstack-br
logdir=c:\openstack\
state_path=c:\openstack\
lock_path=c:\openstack\
instances_path=e:\Hyper-V\instances
policy_file=C:\Program Files (x86)\OpenStack\nova\etc\nova\policy.json
api_paste_config=c:\openstack\nova\etc\nova\api-paste.ini
rabbit_host=IP_ADDRESS
glance_api_servers=IP_ADDRESS:9292
image_service=nova.image.glance.GlanceImageService
instances_shared_storage=false
limit_cpu_features=true
compute_driver=nova.virt.hyperv.driver.HyperVDriver
volume_api_class=nova.volume.cinder.API
[database]
```

```
connection=mysql://nova:passwd@IP_ADDRESS/nova
```

Table 2.28, “Description of configuration options for hyperv” [215] contains a reference of all options for hyper-v.

Prepare images for use with Hyper-V

Hyper-V currently supports only the VHD file format for virtual machine instances. Detailed instructions for installing virtual machines on Hyper-V can be found here:

<http://technet.microsoft.com/en-us/library/cc772480.aspx>

Once you have successfully created a virtual machine, you can then upload the image to glance using the native glance-client:

```
C:\>glance image-create --name="VM_IMAGE_NAME" --is-public=true --container-format=bare --disk-format=vhd
```

Run Compute with Hyper-V

To start the `nova-compute` service, run this command from a console in the Windows server:

```
C:\>C:\python27\python.exe c:\openstack\nova\bin\nova-compute.py
```

Troubleshoot Hyper-V configuration

- I ran the `nova-manage service list` command from my controller; however, I'm not seeing smiley faces for Hyper-V compute nodes, what do I do?

Verify that you are synchronized with a network time source. Instructions for configuring NTP on your Hyper-V compute node are located [here](#)

Baremetal driver

The baremetal driver is a hypervisor driver for OpenStack Nova Compute. Within the OpenStack framework, it has the same role as the drivers for other hypervisors (libvirt, xen, etc), and yet it is presently unique in that the hardware is not virtualized - there is no hypervisor between the tenants and the physical hardware. It exposes hardware through the OpenStack APIs, using pluggable sub-drivers to deliver machine imaging (PXE) and power control (IPMI). With this, provisioning and management of physical hardware is accomplished by using common cloud APIs and tools, such as the Orchestration module (heat) or salt-cloud. However, due to this unique situation, using the baremetal driver requires some additional preparation of its environment, the details of which are beyond the scope of this guide.



Note

Some OpenStack Compute features are not implemented by the baremetal hypervisor driver. See the [hypervisor support matrix](#) for details.

For the Baremetal driver to be loaded and function properly, ensure that the following options are set in `/etc/nova/nova.conf` on your `nova-compute` hosts.

```
[default]
compute_driver=nova.virt.baremetal.driver.BareMetalDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver
scheduler_host_manager=nova.scheduler.baremetal_host_manager.
BaremetalHostManager
ram_allocation_ratio=1.0
reserved_host_memory_mb=0
```

Many configuration options are specific to the Baremetal driver. Also, some additional steps are required, such as building the baremetal deploy ramdisk. See the [main wiki page](#) for details and implementation suggestions.

To customize the Baremetal driver, use the configuration option settings documented in [Table 2.16, "Description of configuration options for baremetal" \[207\]](#).

Scheduling

Compute uses the `nova-scheduler` service to determine how to dispatch compute and volume requests. For example, the `nova-scheduler` service determines which host a VM should launch on. The term *host* in the context of filters means a physical node that has a `nova-compute` service running on it. You can configure the scheduler through a variety of options.

Compute is configured with the following default scheduler options in the `/etc/nova/nova.conf` file:

```
scheduler_driver=nova.scheduler.multi.MultiScheduler
scheduler_driver_task_period=60
compute_scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler
scheduler_available_filters=nova.scheduler.filters.all_filters
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,
ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,
ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter
```

By default, the `scheduler_driver` is configured as a filter scheduler, as described in the next section. In the default configuration, this scheduler considers hosts that meet all the following criteria:

- Have not been attempted for scheduling purposes (`RetryFilter`).
- Are in the requested availability zone (`AvailabilityZoneFilter`).
- Have sufficient RAM available (`RamFilter`).
- Are capable of servicing the request (`ComputeFilter`).
- Satisfy the extra specs associated with the instance type (`ComputeCapabilitiesFilter`).
- Satisfy any architecture, hypervisor type, or virtual machine mode properties specified on the instance's image properties. (`ImagePropertiesFilter`).

The scheduler caches its list of available hosts; you can specify how often the list is updated by modifying the `scheduler_driver_task_period` value.



Note

Do not configure `service_down_time` to be much smaller than `scheduler_driver_task_period`; otherwise, hosts will appear to be dead while the host list is being cached.

For information on the volume scheduler, refer the Block Storage section of [OpenStack Cloud Administrator Guide](#) for information.

The choice of a new host on instance migration is done by the scheduler.

When evacuating instances from a host, the scheduler service does not pick the next host. Instances are evacuated to the host explicitly defined by the administrator. For information about instance evacuation, refer to the [Evacuate instances](#) section of the *Cloud Administrator Guide*.

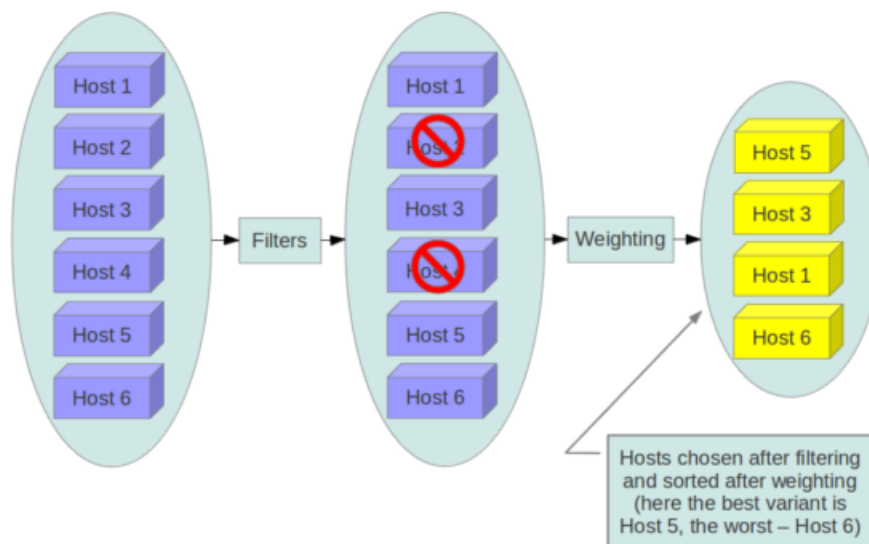
Filter scheduler

The Filter Scheduler (`nova.scheduler.filter_scheduler.FilterScheduler`) is the default scheduler for scheduling virtual machine instances. It supports filtering and weighting to make informed decisions on where a new instance should be created.

Filters

When the Filter Scheduler receives a request for a resource, it first applies filters to determine which hosts are eligible for consideration when dispatching a resource. Filters are binary: either a host is accepted by the filter, or it is rejected. Hosts that are accepted by the filter are then processed by a different algorithm to decide which hosts to use for that request, described in the [Weights](#) section.

Figure 2.2. Filtering



The `scheduler_available_filters` configuration option in `nova.conf` provides the Compute service with the list of the filters that are used by the scheduler. The default setting specifies all of the filter that are included with the Compute service:

```
scheduler_available_filters = nova.scheduler.filters.all_filters
```

This configuration option can be specified multiple times. For example, if you implemented your own custom filter in Python called `myfilter.MyFilter` and you wanted to use both the built-in filters and your custom filter, your `nova.conf` file would contain:

```
scheduler_available_filters=nova.scheduler.filters.all_filters
scheduler_available_filters=myfilter.MyFilter
```

The `scheduler_default_filters` configuration option in `nova.conf` defines the list of filters that are applied by the `nova-scheduler` service. The default filters are:

```
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,
ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,
ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter
```

The following sections describe the available filters.

AggregateCoreFilter

Implements blueprint per-aggregate-resource-ratio. `AggregateCoreFilter` supports per-aggregate `cpu_allocation_ratio`. If the per-aggregate value is not found, the value falls back to the global setting.

AggregateImagePropertiesIsolation

Matches properties defined in an image's metadata against those of aggregates to determine host matches:

- If a host belongs to an aggregate and the aggregate defines one or more metadata that match an image's properties, that host is a candidate to boot the image's instance.
- If a host does not belong to any aggregate, it can boot instances from all images.

For example, the following aggregate `myWinAgg` has the Windows operating system as metadata (named 'windows'):

```
$ nova aggregate-details MyWinAgg
+-----+-----+-----+-----+-----+
| Id | Name      | Availability Zone | Hosts      | Metadata      |
+-----+-----+-----+-----+-----+
| 1  | MyWinAgg | None              | 'sf-devel' | 'os=windows'  |
+-----+-----+-----+-----+-----+
```

In this example, because the following Win-2012 image has the windows property, it would boot on the `sf-devel` host (all other filters being equal):

```
$ glance image-show Win-2012
+-----+-----+
| Property      | Value      |
+-----+-----+
| Property 'os' | windows    |
| checksum      | f8a2eeee2dc65b3d9b6e63678955bd83 |
+-----+-----+
```

```
| container_format | ami |
| created_at       | 2013-11-14T13:24:25 |
| ...
```

You can configure the `AggregateImagePropertiesIsolation` filter using the following options in the `nova.conf` file:

```
# Considers only keys matching the given namespace (string).
aggregate_image_properties_isolation_namespace=<None>

# Separator used between the namespace and keys (string).
aggregate_image_properties_isolation_separator=.
```

AggregateInstanceExtraSpecsFilter

Matches properties defined in an instance type's extra specs against admin-defined properties on a host aggregate. Works with specifications that are unscoped, or are scoped with `aggregate_instance_extra_specs`. See the [host aggregates](#) section for documentation on how to use this filter.

AggregateMultiTenancyIsolation

Isolates tenants to specific [host aggregates](#). If a host is in an aggregate that has the metadata key `filter_tenant_id` it only creates instances from that tenant (or list of tenants). A host can be in different aggregates. If a host does not belong to an aggregate with the metadata key, it can create instances from all tenants.

AggregateRamFilter

Implements blueprint `per-aggregate-resource-ratio`. Supports per-aggregate `ram_allocation_ratio`. If per-aggregate value is not found, it falls back to the default setting.

AllHostsFilter

This is a no-op filter, it does not eliminate any of the available hosts.

AvailabilityZoneFilter

Filters hosts by availability zone. This filter must be enabled for the scheduler to respect availability zones in requests.

ComputeCapabilitiesFilter

Matches properties defined in an instance type's extra specs against compute capabilities.

If an extra specs key contains a colon ":", anything before the colon is treated as a namespace, and anything after the colon is treated as the key to be matched. If a namespace is present and is not 'capabilities', it is ignored by this filter.

ComputeFilter

Passes all hosts that are operational and enabled.

In general, this filter should always be enabled.

CoreFilter

Only schedule instances on hosts if there are sufficient CPU cores available. If this filter is not set, the scheduler may over provision a host based on cores (for example, the virtual cores running on an instance may exceed the physical cores).

This filter can be configured to allow a fixed amount of vCPU overcommitment by using the `cpu_allocation_ratio` Configuration option in `nova.conf`. The default setting is:

```
cpu_allocation_ratio=16.0
```

With this setting, if 8 vCPUs are on a node, the scheduler allows instances up to 128 vCPU to be run on that node.

To disallow vCPU overcommitment set:

```
cpu_allocation_ratio=1.0
```



Note

The Compute API will always return the actual number of CPU cores available on a compute node regardless of the value of the `cpu_allocation_ratio` configuration key. As a result changes to the `cpu_allocation_ratio` are not reflected via the command line clients or the dashboard. Changes to this configuration key are only taken into account internally in the scheduler.

DifferentHostFilter

Schedule the instance on a different host from a set of instances. To take advantage of this filter, the requester must pass a scheduler hint, using `different_host` as the key and a list of instance uuids as the value. This filter is the opposite of the `SameHostFilter`. Using the `nova` command-line tool, use the `--hint` flag. For example:

```
$ nova boot --image cedef40a-ed67-4d10-800e-17455edce175 --flavor 1 \  
--hint different_host=a0cf03a5-d921-4877-bb5c-86d26cf818e1 \  
--hint different_host=8c19174f-4220-44f0-824a-cd1eeef10287 server-1
```

With the API, use the `os:scheduler_hints` key. For example:

```
{  
  "server": {  
    "name": "server-1",  
    "imageRef": "cedef40a-ed67-4d10-800e-17455edce175",  
    "flavorRef": "1"  
  },  
  "os:scheduler_hints": {  
    "different_host": [  
      "a0cf03a5-d921-4877-bb5c-86d26cf818e1",  
      "8c19174f-4220-44f0-824a-cd1eeef10287"  
    ]  
  }  
}
```

DiskFilter

Only schedule instances on hosts if there is sufficient disk space available for root and ephemeral storage.

This filter can be configured to allow a fixed amount of disk overcommitment by using the `disk_allocation_ratio` Configuration option in `nova.conf`. The default setting is:

```
disk_allocation_ratio=1.0
```

Adjusting this value to greater than 1.0 enables scheduling instances while over committing disk resources on the node. This might be desirable if you use an image format that is sparse or copy on write such that each virtual instance does not require a 1:1 allocation of virtual disk to physical storage.

GroupAffinityFilter



Note

This filter is deprecated in favor of [ServerGroupAffinityFilter](#).

The `GroupAffinityFilter` ensures that an instance is scheduled on to a host from a set of group hosts. To take advantage of this filter, the requester must pass a scheduler hint, using `group` as the key and an arbitrary name as the value. Using the `nova` command-line tool, use the `--hint` flag. For example:

```
$ nova boot --image IMAGE_ID --flavor 1 --hint group=foo server-1
```

This filter should not be enabled at the same time as [GroupAntiAffinityFilter](#) or neither filter will work properly.

GroupAntiAffinityFilter



Note

This filter is deprecated in favor of [ServerGroupAntiAffinityFilter](#).

The `GroupAntiAffinityFilter` ensures that each instance in a group is on a different host. To take advantage of this filter, the requester must pass a scheduler hint, using `group` as the key and an arbitrary name as the value. Using the `nova` command-line tool, use the `--hint` flag. For example:

```
$ nova boot --image IMAGE_ID --flavor 1 --hint group=foo server-1
```

This filter should not be enabled at the same time as [GroupAffinityFilter](#) or neither filter will work properly.

ImagePropertiesFilter

Filters hosts based on properties defined on the instance's image. It passes hosts that can support the specified image properties contained in the instance. Properties include the

architecture, hypervisor type, and virtual machine mode. for example, an instance might require a host that runs an ARM-based processor and QEMU as the hypervisor. An image can be decorated with these properties by using:

```
$ glance image-update img-uuid --property architecture=arm --property  
  hypervisor_type=qemu
```

The image properties that the filter checks for are:

- **architecture:** Architecture describes the machine architecture required by the image. Examples are `i686`, `x86_64`, `arm`, and `ppc64`.
- **hypervisor_type:** Hypervisor type describes the hypervisor required by the image. Examples are `xen`, `kvm`, `qemu`, and `xenapi`.
- **vm_mode:** Virtual machine mode describes the hypervisor application binary interface (ABI) required by the image. Examples are 'xen' for Xen 3.0 paravirtual ABI, 'hvm' for native ABI, 'uml' for User Mode Linux paravirtual ABI, `exe` for container virt executable ABI.

IsolatedHostsFilter

Allows the admin to define a special (isolated) set of images and a special (isolated) set of hosts, such that the isolated images can only run on the isolated hosts, and the isolated hosts can only run isolated images. The flag `restrict_isolated_hosts_to_isolated_images` can be used to force isolated hosts to only run isolated images.

The admin must specify the isolated set of images and hosts in the `nova.conf` file using the `isolated_hosts` and `isolated_images` configuration options. For example:

```
isolated_hosts=server1,server2  
isolated_images=342b492c-128f-4a42-8d3a-c5088cf27d13,ebd267a6-ca86-4d6c-9a0e-  
bd132d6b7d09
```

JsonFilter

The `JsonFilter` allows a user to construct a custom filter by passing a scheduler hint in JSON format. The following operators are supported:

- `=`
- `<`
- `>`
- `in`
- `<=`
- `>=`
- `not`

- or
- and

The filter supports the following variables:

- `$free_ram_mb`
- `$free_disk_mb`
- `$total_usable_ram_mb`
- `$vcpus_total`
- `$vcpus_used`

Using the `nova` command-line tool, use the `--hint` flag:

```
$ nova boot --image 827d564a-e636-4fc4-a376-d36f7ebe1747 \  
--flavor 1 --hint query='[">=", "$free_ram_mb", 1024]' server1
```

With the API, use the `os:scheduler_hints` key:

```
{  
  "server": {  
    "name": "server-1",  
    "imageRef": "cedef40a-ed67-4d10-800e-17455edce175",  
    "flavorRef": "1"  
  },  
  "os:scheduler_hints": {  
    "query": "[>=, $free_ram_mb, 1024]"  
  }  
}
```

RamFilter

Only schedule instances on hosts that have sufficient RAM available. If this filter is not set, the scheduler may over provision a host based on RAM (for example, the RAM allocated by virtual machine instances may exceed the physical RAM).

This filter can be configured to allow a fixed amount of RAM overcommitment by using the `ram_allocation_ratio` configuration option in `nova.conf`. The default setting is:

```
ram_allocation_ratio=1.5
```

This setting enables 1.5 GB instances to run on any compute node with 1 GB of free RAM.

RetryFilter

Filter out hosts that have already been attempted for scheduling purposes. If the scheduler selects a host to respond to a service request, and the host fails to respond to the request, this filter prevents the scheduler from retrying that host for the service request.

This filter is only useful if the `scheduler_max_attempts` configuration option is set to a value greater than zero.

SameHostFilter

Schedule the instance on the same host as another instance in a set of instances. To take advantage of this filter, the requester must pass a scheduler hint, using `same_host` as the key and a list of instance uuids as the value. This filter is the opposite of the `DifferentHostFilter`. Using the `nova` command-line tool, use the `--hint` flag:

```
$ nova boot --image cedef40a-ed67-4d10-800e-17455edce175 --flavor 1 \  
--hint same_host=a0cf03a5-d921-4877-bb5c-86d26cf818e1 \  
--hint same_host=8c19174f-4220-44f0-824a-cd1eeef10287 server-1
```

With the API, use the `os:scheduler_hints` key:

```
{  
  "server": {  
    "name": "server-1",  
    "imageRef": "cedef40a-ed67-4d10-800e-17455edce175",  
    "flavorRef": "1"  
  },  
  "os:scheduler_hints": {  
    "same_host": [  
      "a0cf03a5-d921-4877-bb5c-86d26cf818e1",  
      "8c19174f-4220-44f0-824a-cd1eeef10287"  
    ]  
  }  
}
```

ServerGroupAffinityFilter

The `ServerGroupAffinityFilter` ensures that an instance is scheduled on to a host from a set of group hosts. To take advantage of this filter, the requester must create a server group with an affinity policy, and pass a scheduler hint, using `group` as the key and the server group UUID as the value. Using the `nova` command-line tool, use the `--hint` flag. For example:

```
$ nova server-group-create --policy affinity group-1  
$ nova boot --image IMAGE_ID --flavor 1 --hint group=SERVER_GROUP_UUID  
server-1
```

ServerGroupAntiAffinityFilter

The `ServerGroupAntiAffinityFilter` ensures that each instance in a group is on a different host. To take advantage of this filter, the requester must create a server group with an anti-affinity policy, and pass a scheduler hint, using `group` as the key and the server group UUID as the value. Using the `nova` command-line tool, use the `--hint` flag. For example:

```
$ nova server-group-create --policy anti-affinity group-1  
$ nova boot --image IMAGE_ID --flavor 1 --hint group=SERVER_GROUP_UUID  
server-1
```

SimpleCIDRAffinityFilter

Schedule the instance based on host IP subnet range. To take advantage of this filter, the requester must specify a range of valid IP address in CIDR format, by passing two scheduler hints:

`build_near_host_ip` The first IP address in the subnet (for example, 192.168.1.1)

`cidr` The CIDR that corresponds to the subnet (for example, /24)

Using the **nova** command-line tool, use the `--hint` flag. For example, to specify the IP subnet 192.168.1.1/24

```
$ nova boot --image cedef40a-ed67-4d10-800e-17455edce175 --flavor 1 \  
--hint build_near_host_ip=192.168.1.1 --hint cidr=/24 server-1
```

With the API, use the `os:scheduler_hints` key:

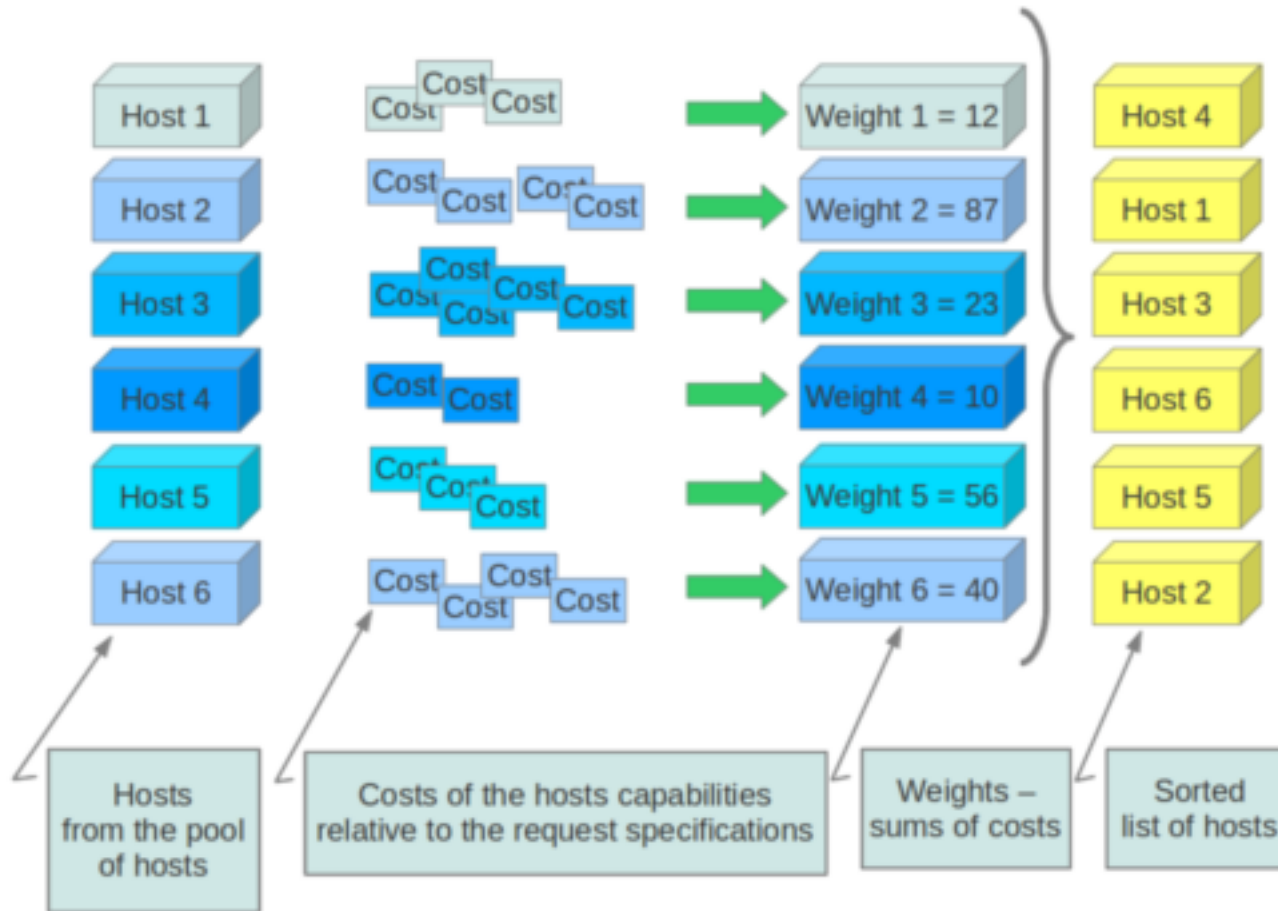
```
{  
  "server": {  
    "name": "server-1",  
    "imageRef": "cedef40a-ed67-4d10-800e-17455edce175",  
    "flavorRef": "1"  
  },  
  "os:scheduler_hints": {  
    "build_near_host_ip": "192.168.1.1",  
    "cidr": "24"  
  }  
}
```

Weights

When resourcing instances, the Filter Scheduler filters and weighs each host in the list of acceptable hosts. Each time the scheduler selects a host, it virtually consumes resources on it, and subsequent selections are adjusted accordingly. This process is useful when the customer asks for the same large amount of instances, because weight is computed for each requested instance.

All weights are normalized before being summed up; the host with the largest weight is given the highest priority.

Figure 2.3. Weighing hosts



If cells are used, cells are weighed by the scheduler in the same manner as hosts.

Hosts and cells are weighed based on the following options in the `/etc/nova/nova.conf` file:

Table 2.8. Host Weighting options

Section	Option	Description
[DEFAULT]	<code>ram_weight_multiplier</code>	By default, the scheduler spreads instances across all hosts evenly. Set the <code>ram_weight_multiplier</code> option to a negative number if you prefer stacking instead of spreading. Use a floating-point value.
[DEFAULT]	<code>scheduler_host_subset</code>	New instances are scheduled on a host that is chosen randomly from a subset of the N best hosts. This property defines the subset size from which a host is chosen. A value of 1 chooses the first host returned by the weighing functions. This value must be at least 1. A value less than 1 is ignored, and 1 is used instead. Use an integer value.
[DEFAULT]	<code>scheduler_weight_class</code>	Defaults to <code>nova.scheduler.weights.all_weighters</code> , which selects the only available weigher, the <code>RamWeigher</code> . Hosts are then weighed and sorted with the largest weight winning.
[metrics]	<code>weight_multiplier</code>	Multiplier for weighing metrics. Use a floating-point value.

Section	Option	Description
[metrics]	weight_setting	Determines how metrics are weighed. Use a comma-separated list of <code>metricName=ratio</code> . For example: "name1=1.0, name2=-1.0" results in: $name1.value * 1.0 + name2.value * -1.0$
[metrics]	required	Specifies how to treat unavailable metrics: <ul style="list-style-type: none"> • True—Raises an exception. To avoid the raised exception, you should use the scheduler filter <code>MetricFilter</code> to filter out hosts with unavailable metrics. • False—Treated as a negative factor in the weighing process (uses the <code>weight_of_unavailable</code> option).
[metrics]	weight_of_unavailable	If <code>required</code> is set to False, and any one of the metrics set by <code>weight_setting</code> is unavailable, the <code>weight_of_unavailable</code> value is returned to the scheduler.

For example:

```
[DEFAULT]
scheduler_host_subset_size=1
scheduler_weight_classes=nova.scheduler.weights.all_weighers
ram_weight_multiplier=1.0
[metrics]
weight_multiplier=1.0
weight_setting=name1=1.0, name2=-1.0
required=false
weight_of_unavailable=-10000.0
```

Table 2.9. Cell weighting options

Section	Option	Description
[cells]	mute_weight_multiplier	Multiplier to weigh mute children (hosts which have not sent capacity or capacity updates for some time). Use a negative, floating-point value.
[cells]	mute_weight_value	Weight value assigned to mute children. Use a positive, floating-point value with a maximum of '1.0'.
[cells]	offset_weight_multiplier	Multiplier to weigh cells, so you can specify a preferred cell. Use a floating point value.
[cells]	ram_weight_multiplier	By default, the scheduler spreads instances across all cells evenly. Set the <code>ram_weight_multiplier</code> option to a negative number if you prefer stacking instead of spreading. Use a floating-point value.
[cells]	scheduler_weight_classes	Defaults to <code>nova.cells.weights.all_weighers</code> , which maps to all cell weighers included with Compute. Cells are then weighed and sorted with the largest weight winning.

For example:

```
[cells]
scheduler_weight_classes=nova.cells.weights.all_weighers
mute_weight_multiplier=-10.0
mute_weight_value=1000.0
ram_weight_multiplier=1.0
offset_weight_multiplier=1.0
```

Chance scheduler

As an administrator, you work with the Filter Scheduler. However, the Compute service also uses the Chance Scheduler, `nova.scheduler.chance.ChanceScheduler`, which randomly selects from lists of filtered hosts.

Host aggregates

Host aggregates are a mechanism to further partition an availability zone; while availability zones are visible to users, host aggregates are only visible to administrators. Host Aggregates provide a mechanism to allow administrators to assign key-value pairs to groups of machines. Each node can have multiple aggregates, each aggregate can have multiple key-value pairs, and the same key-value pair can be assigned to multiple aggregates. This information can be used in the scheduler to enable advanced scheduling, to set up hypervisor resource pools or to define logical groups for migration.

Command-line interface

The `nova` command-line tool supports the following aggregate-related commands.

<code>nova aggregate-list</code>	Print a list of all aggregates.
<code>nova aggregate-create <name> <availability-zone></code>	Create a new aggregate named <code><name></code> in availability zone <code><availability-zone></code> . Returns the ID of the newly created aggregate. Hosts can be made available to multiple availability zones, but administrators should be careful when adding the host to a different host aggregate within the same availability zone and pay attention when using the <code>aggregate-set-metadata</code> and <code>aggregate-update</code> commands to avoid user confusion when they boot instances in different availability zones. An error occurs if you cannot add a particular host to an aggregate zone for which it is not intended.
<code>nova aggregate-delete <id></code>	Delete an aggregate with id <code><id></code> .
<code>nova aggregate-details <id></code>	Show details of the aggregate with id <code><id></code> .
<code>nova aggregate-add-host <id> <host></code>	Add host with name <code><host></code> to aggregate with id <code><id></code> .
<code>nova aggregate-remove-host <id> <host></code>	Remove the host with name <code><host></code> from the aggregate with id <code><id></code> .
<code>nova aggregate-set-metadata <id> <key=value> [<key=value> ...]</code>	Add or update metadata (key-value pairs) associated with the aggregate with id <code><id></code> .
<code>nova aggregate- update <id> <name> [<availability_zone>]</code>	Update the name and availability zone (optional) for the aggregate.
<code>nova host-list</code>	List all hosts by service.
<code>nova host-update - maintenance [enable disable]</code>	Put/resume host into/from maintenance.



Note

Only administrators can access these commands. If you try to use these commands and the user name and tenant that you use to access the Compute service do not have the `admin` role or the appropriate privileges, these errors occur:

```
ERROR: Policy doesn't allow compute_extension:aggregates to be
performed. (HTTP 403) (Request-ID: req-299fbff6-6729-4cef-93b2-
e7e1f96b4864)
```

```
ERROR: Policy doesn't allow compute_extension:hosts to be performed.
(HTTP 403) (Request-ID: req-ef2400f6-6776-4ea3-b6f1-7704085c27d1)
```

Configure scheduler to support host aggregates

One common use case for host aggregates is when you want to support scheduling instances to a subset of compute hosts because they have a specific capability. For example, you may want to allow users to request compute hosts that have SSD drives if they need access to faster disk I/O, or access to compute hosts that have GPU cards to take advantage of GPU-accelerated code.

To configure the scheduler to support host aggregates, the `scheduler_default_filters` configuration option must contain the `AggregateInstanceExtraSpecsFilter` in addition to the other filters used by the scheduler. Add the following line to `/etc/nova/nova.conf` on the host that runs the `nova-scheduler` service to enable host aggregates filtering, as well as the other filters that are typically enabled:

```
scheduler_default_filters=AggregateInstanceExtraSpecsFilter,
AvailabilityZoneFilter,RamFilter,ComputeFilter
```

Example: Specify compute hosts with SSDs

This example configures the Compute service to enable users to request nodes that have solid-state drives (SSDs). You create a `fast-io` host aggregate in the `nova` availability zone and you add the `ssd=true` key-value pair to the aggregate. Then, you add the `node1`, and `node2` compute nodes to it.

```
$ nova aggregate-create fast-io nova
+-----+-----+-----+-----+-----+
| Id | Name   | Availability Zone | Hosts | Metadata |
+-----+-----+-----+-----+-----+
| 1  | fast-io | nova              |      |          |
+-----+-----+-----+-----+-----+

$ nova aggregate-set-metadata 1 ssd=true
+-----+-----+-----+-----+-----+
| Id | Name   | Availability Zone | Hosts | Metadata          |
+-----+-----+-----+-----+-----+
| 1  | fast-io | nova              | []    | {'ssd': 'true'}  |
+-----+-----+-----+-----+-----+

$ nova aggregate-add-host 1 node1
+-----+-----+-----+-----+-----+
| Id | Name   | Availability Zone | Hosts | Metadata          |
+-----+-----+-----+-----+-----+
```


XenServer hypervisor pools to support live migration

When using the XenAPI-based hypervisor, the Compute service uses host aggregates to manage XenServer Resource pools, which are used in supporting live migration.

Configuration reference

To customize the Compute scheduler, use the configuration option settings documented in [Table 2.48, “Description of configuration options for scheduling” \[227\]](#).

Cells

Cells functionality allows you to scale an OpenStack Compute cloud in a more distributed fashion without having to use complicated technologies like database and message queue clustering. It is intended to support very large deployments.

When this functionality is enabled, the hosts in an OpenStack Compute cloud are partitioned into groups called cells. Cells are configured as a tree. The top-level cell should have a host that runs a `nova-api` service, but no `nova-compute` services. Each child cell should run all of the typical `nova-*` services in a regular Compute cloud except for `nova-api`. You can think of cells as a normal Compute deployment in that each cell has its own database server and message queue broker.

The `nova-cells` service handles communication between cells and selects cells for new instances. This service is required for every cell. Communication between cells is pluggable, and currently the only option is communication through RPC.

Cells scheduling is separate from host scheduling. `nova-cells` first picks a cell. Once a cell is selected and the new build request reaches its `nova-cells` service, it is sent over to the host scheduler in that cell and the build proceeds as it would have without cells.



Warning

Cell functionality is currently considered experimental.

Cell configuration options

Cells are disabled by default. All cell-related configuration options go under a `[cells]` section in `nova.conf`. The following cell-related options are currently supported:

<code>enable</code>	Set this to <code>True</code> to turn on cell functionality, which is off by default.
<code>name</code>	Name of the current cell. This must be unique for each cell.
<code>capabilities</code>	List of arbitrary <code>key=value</code> pairs defining capabilities of the current cell. Values include <code>hypervisor=xenserver;kvm,os=linux;windows</code> .
<code>call_timeout</code>	How long in seconds to wait for replies from calls between cells.


```
--name=<name>           Name for the new cell
--cell_type=<parent|child>
                        Whether the cell is a parent or child
--username=<username>   Username for the message broker in this cell
--password=<password>   Password for the message broker in this cell
--hostname=<hostname>   Address of the message broker in this cell
--port=<number>         Port number of the message broker in this cell
--virtual_host=<virtual_host>
                        The virtual host of the message broker in this cell
--woffset=<float>
                        (weight offset) It might be used by some cell
                        scheduling code in the future
--wscale=<float>
                        (weight scale) It might be used by some cell
                        scheduling code in the future
```

As an example, assume we have an API cell named `api` and a child cell named `cell1`. Within the `api` cell, we have the following RabbitMQ server info:

```
rabbit_host=10.0.0.10
rabbit_port=5672
rabbit_username=api_user
rabbit_password=api_passwd
rabbit_virtual_host=api_vhost
```

And in the child cell named `cell1` we have the following RabbitMQ server info:

```
rabbit_host=10.0.1.10
rabbit_port=5673
rabbit_username=cell1_user
rabbit_password=cell1_passwd
rabbit_virtual_host=cell1_vhost
```

We would run this in the API cell, as root.

```
# nova-manage cell create --name=cell1 --cell_type=child \
  --username=cell1_user --password=cell1_passwd --hostname=10.0.1.10 \
  --port=5673 --virtual_host=cell1_vhost --woffset=1.0 --wscale=1.0
```

Repeat the above for all child cells.

In the child cell, we would run the following, as root:

```
# nova-manage cell create --name=api --cell_type=parent \
  --username=api_user --password=api_passwd --hostname=10.0.0.10 \
  --port=5672 --virtual_host=api_vhost --woffset=1.0 --wscale=1.0
```

To customize the Compute cells, use the configuration option settings documented in [Table 2.18, "Description of configuration options for cells" \[209\]](#).

Cell scheduling configuration

To determine the best cell for launching a new instance, Compute uses a set of filters and weights configured in `/etc/nova/nova.conf`. The following options are available to prioritize cells for scheduling:

- `scheduler_filter_classes` - Specifies the list of filter classes. By default `nova.cells.weights.all_filters` is specified, which maps to all cells filters included with Compute (see [the section called "Filters" \[181\]](#)).
- `scheduler_weight_classes` - Specifies the list of weight classes. By default `nova.cells.weights.all_weighters` is specified, which maps to all cell weight algorithms (weighters) included with Compute. The following modules are available:
 - `mute_child`: Downgrades the likelihood of child cells being chosen for scheduling requests, which haven't sent capacity or capability updates in a while. Options include `mute_weight_multiplier` (multiplier for mute children; value should be negative) and `mute_weight_value` (assigned to mute children; should be a positive value).
 - `ram_by_instance_type`: Select cells with the most RAM capacity for the instance type being requested. Because higher weights win, Compute returns the number of available units for the instance type requested. The `ram_weight_multiplier` option defaults to 10.0 that adds to the weight by a factor of 10. Use a negative number to stack VMs on one host instead of spreading out new VMs to more hosts in the cell.
 - `weight_offset`: Allows modifying the database to weight a particular cell. You can use this when you want to disable a cell (for example, '0'), or to set a default cell by making its `weight_offset` very high (for example, '9999999999999999'). The highest weight will be the first cell to be scheduled for launching an instance.

Additionally, the following options are available for the cell scheduler:

- `scheduler_retries` - Specifies how many times the scheduler tries to launch a new instance when no cells are available (default=10).
- `scheduler_retry_delay` - Specifies the delay (in seconds) between retries (default=2).

As an admin user, you can also add a filter that directs builds to a particular cell. The `policy.json` file must have a line with `"cells_scheduler_filter:TargetCellFilter" : "is_admin:True"` to let an admin user specify a scheduler hint to direct a build to a particular cell.

Optional cell configuration

Cells currently keeps all inter-cell communication data, including user names and passwords, in the database. This is undesirable and unnecessary since cells data isn't updated very frequently. Instead, create a JSON file to input cells data specified via a `[cells]cells_config` option. When specified, the database is no longer consulted when reloading cells data. The file will need the columns present in the Cell model (excluding common database fields and the `id` column). The queue connection information must be specified through a `transport_url` field, instead of `username`, `password`, and so on. The `transport_url` has the following form:

```
rabbit://USERNAME:PASSWORD@HOSTNAME:PORT/VIRTUAL_HOST
```

The scheme can be either `qpid` or `rabbit`, as shown previously. The following sample shows this optional configuration:

```
{
  "parent": {
    "name": "parent",
    "api_url": "http://api.example.com:8774",
    "transport_url": "rabbit://rabbit.example.com",
    "weight_offset": 0.0,
    "weight_scale": 1.0,
    "is_parent": true
  },
  "cell1": {
    "name": "cell1",
    "api_url": "http://api.example.com:8774",
    "transport_url": "rabbit://rabbit1.example.com",
    "weight_offset": 0.0,
    "weight_scale": 1.0,
    "is_parent": false
  },
  "cell2": {
    "name": "cell2",
    "api_url": "http://api.example.com:8774",
    "transport_url": "rabbit://rabbit2.example.com",
    "weight_offset": 0.0,
    "weight_scale": 1.0,
    "is_parent": false
  }
}
```

Conductor

The `nova-conductor` service enables OpenStack to function without compute nodes accessing the database. Conceptually, it implements a new layer on top of `nova-compute`. It should not be deployed on compute nodes, or else the security benefits of removing database access from `nova-compute` are negated. Just like other nova services such as `nova-api` or `nova-scheduler`, it can be scaled horizontally. You can run multiple instances of `nova-conductor` on different machines as needed for scaling purposes.

The methods exposed by `nova-conductor` are relatively simple methods used by `nova-compute` to offload its database operations. Places where `nova-compute` previously performed database access are now talking to `nova-conductor`. However, we have plans in the medium to long term to move more and more of what is currently in `nova-compute` up to the `nova-conductor` layer. The Compute service will start to look like a less intelligent slave service to `nova-conductor`. The conductor service will implement long running complex operations, ensuring forward progress and graceful error handling. This will be especially beneficial for operations that cross multiple compute nodes, such as migrations or resizes.

To customize the Conductor, use the configuration option settings documented in [Table 2.21, "Description of configuration options for conductor" \[212\]](#).

Example `nova.conf` configuration files

The following sections describe the configuration options in the `nova.conf` file. You must copy the `nova.conf` file to each compute node. The sample `nova.conf` files show examples of specific configurations.

Small, private cloud

This example `nova.conf` file configures a small private cloud with cloud controller services, database server, and messaging server on the same server. In this case, `CONTROLLER_IP` represents the IP address of a central server, `BRIDGE_INTERFACE` represents the bridge such as `br100`, the `NETWORK_INTERFACE` represents an interface to your VLAN setup, and passwords are represented as `DB_PASSWORD_COMPUTE` for your Compute (nova) database password, and `RABBIT_PASSWORD` represents the password to your message queue installation.

```
[DEFAULT]

# LOGS/STATE
verbose=True
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
rootwrap_config=/etc/nova/rootwrap.conf

# SCHEDULER
compute_scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler

# VOLUMES
# configured in cinder.conf

# COMPUTE
compute_driver=libvirt.LibvirtDriver
instance_name_template=instance-%08x
api_paste_config=/etc/nova/api-paste.ini

# COMPUTE/APIS: if you have separate configs for separate services
# this flag is required for both nova-api and nova-compute
allow_resize_to_same_host=True

# APIS
osapi_compute_extension=nova.api.openstack.compute.contrib.standard_extensions
ec2_dmz_host=192.168.206.130
s3_host=192.168.206.130

# RABBITMQ
rabbit_host=192.168.206.130

# GLANCE
image_service=nova.image.glance.GlanceImageService
glance_api_servers=192.168.206.130:9292

# NETWORK
network_manager=nova.network.manager.FlatDHCPManager
force_dhcp_release=True
dhcpbridge_flagfile=/etc/nova/nova.conf
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
# Change my_ip to match each host
my_ip=192.168.206.130
public_interface=eth0
vlan_interface=eth0
flat_network_bridge=br100
flat_interface=eth0
```

```
# NOVNC CONSOLE
novncproxy_base_url=http://192.168.206.130:6080/vnc_auto.html
# Change vncserver_proxyclient_address and vncserver_listen to match each
compute host
vncserver_proxyclient_address=192.168.206.130
vncserver_listen=192.168.206.130

# AUTHENTICATION
auth_strategy=keystone
[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = nova
signing_dirname = /tmp/keystone-signing-nova

# DATABASE
[database]
connection=mysql://nova:yourpassword@192.168.206.130/nova

# LIBVIRT
[libvirt]
virt_type=qemu
```

KVM, Flat, MySQL, and Glance, OpenStack or EC2 API

This example `nova.conf` file, from an internal Rackspace test system, is used for demonstrations.

```
[DEFAULT]

# LOGS/STATE
verbose=True
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
rootwrap_config=/etc/nova/rootwrap.conf

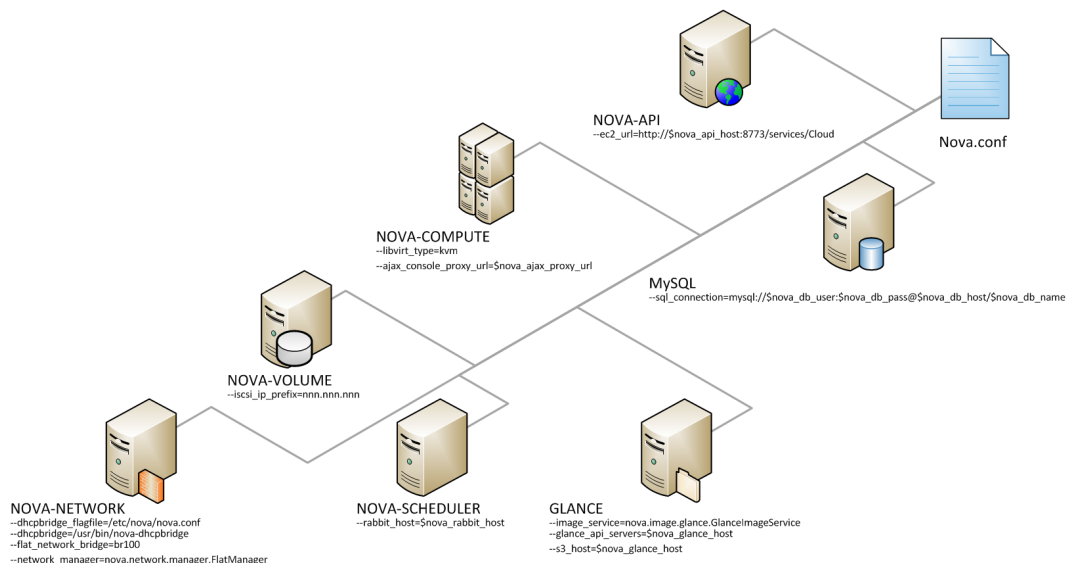
# SCHEDULER
compute_scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler

# VOLUMES
# configured in cinder.conf

# COMPUTE
compute_driver=libvirt.LibvirtDriver
instance_name_template=instance-%08x
api_paste_config=/etc/nova/api-paste.ini

# COMPUTE/APIS: if you have separate configs for separate services
# this flag is required for both nova-api and nova-compute
allow_resize_to_same_host=True

# APIS
osapi_compute_extension=nova.api.openstack.compute.contrib.standard_extensions
ec2_dmz_host=192.168.206.130
s3_host=192.168.206.130
```


Figure 2.4. KVM, Flat, MySQL, and Glance, OpenStack or EC2 API

XenServer, Flat networking, MySQL, and Glance, OpenStack API

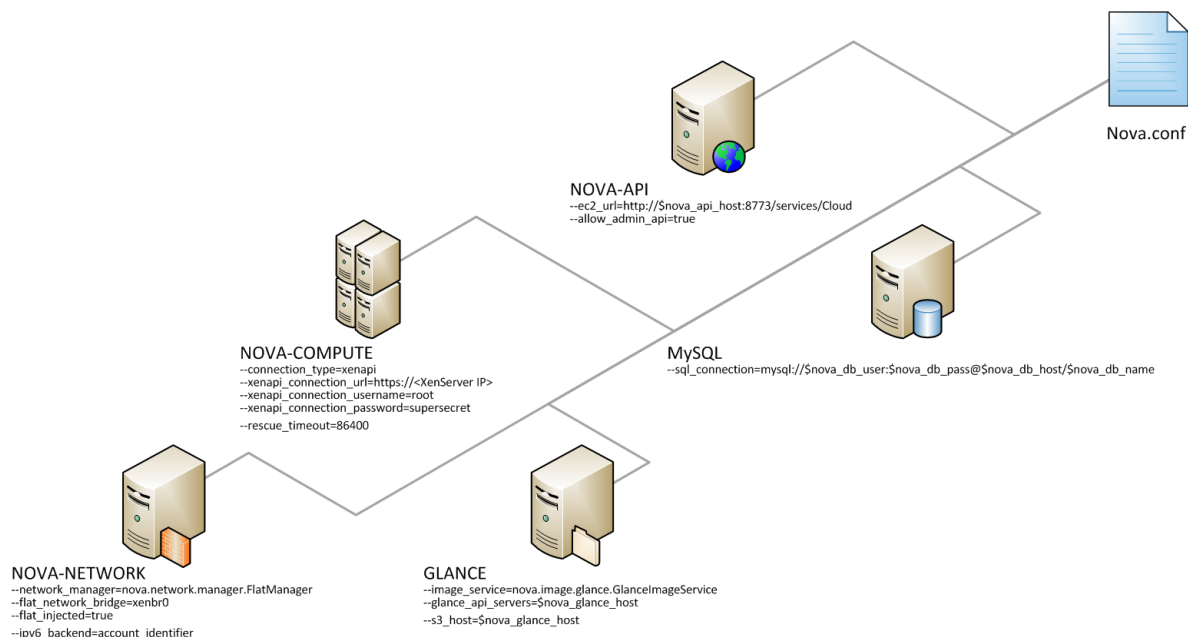
This example `nova.conf` file is from an internal Rackspace test system.

```
verbose
nodaemon
network_manager=nova.network.manager.FlatManager
image_service=nova.image.glance.GlanceImageService
flat_network_bridge=xenbr0
compute_driver=xenapi.XenAPIDriver
xenapi_connection_url=https://<XenServer IP>
xenapi_connection_username=root
xenapi_connection_password=supersecret
xenapi_image_upload_handler=nova.virt.xenapi.image.glance.GlanceStore
rescue_timeout=86400
use_ipv6=true

# To enable flat_injected, currently only works on Debian-based systems
flat_injected=true
ipv6_backend=account_identifier
ca_path=./nova/CA

# Add the following to your conf file if you're running on Ubuntu Maverick
xenapi_remap_vbd_dev=true
[database]
connection=mysql://root:<password>@127.0.0.1/nova
```

Figure 2.5. KVM, Flat, MySQL, and Glance, OpenStack or EC2 API



Compute log files

The corresponding log file of each Compute service is stored in the `/var/log/nova/` directory of the host on which each service runs.

Table 2.10. Log files used by Compute services

Log file	Service name (CentOS/Fedora/ openSUSE/Red Hat Enterprise Linux/ SUSE Linux Enterprise)	Service name (Ubuntu/Debian)
<code>api.log</code>	<code>openstack-nova-api</code>	<code>nova-api</code>
<code>cert.log</code> ^a	<code>openstack-nova-cert</code>	<code>nova-cert</code>
<code>compute.log</code>	<code>openstack-nova-compute</code>	<code>nova-compute</code>
<code>conductor.log</code>	<code>openstack-nova-conductor</code>	<code>nova-conductor</code>
<code>consoleauth.log</code>	<code>openstack-nova-consoleauth</code>	<code>nova-consoleauth</code>
<code>network.log</code> ^b	<code>openstack-nova-network</code>	<code>nova-network</code>
<code>nova-manage.log</code>	<code>nova-manage</code>	<code>nova-manage</code>
<code>scheduler.log</code>	<code>openstack-nova-scheduler</code>	<code>nova-scheduler</code>

^aThe X509 certificate service (`openstack-nova-cert/nova-cert`) is only required by the EC2 API to the Compute service.

^bThe nova network service (openstack-nova-network/nova-network) only runs in deployments that are not configured to use the Networking service (neutron).

Compute sample configuration files

nova.conf - configuration options

For a complete list of all available configuration options for each OpenStack Compute service, run `bin/nova-<servicename> -help`.

Table 2.11. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
api_rate_limit = False	(BoolOpt) Whether to use per-user rate limiting for the api. This option is only used by v2 api. Rate limiting is removed from v3 api.
enable_new_services = True	(BoolOpt) Services to be added to the available pool on create
enabled_apis = ec2, osapi_compute, metadata	(ListOpt) A list of APIs to enable by default
enabled_ssl_apis =	(ListOpt) A list of APIs with enabled SSL
instance_name_template = instance-%08x	(StrOpt) Template string to be used to generate instance names
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
multi_instance_display_name_template = %(name)s-%(uuid)s	(StrOpt) When creating multiple instances with a single request using the os-multiple-create API extension, this template will be used to build the display name for each instance. The benefit is that the instances end up with different hostnames. To restore legacy behavior of every instance having the same name, set this option to "%(name)s". Valid keys for the template are: name, uuid, count.
non_inheritable_image_properties = cache_in_nova, bittorrent	(ListOpt) These are image properties which a snapshot should not inherit from an instance
null_kernel = nokernel	(StrOpt) Kernel image that indicates not to use a kernel, but to use a raw disk image instead
osapi_compute_ext_list =	(ListOpt) Specify list of extensions to load when using osapi_compute_extension option with nova.api.openstack.compute.contrib.select_extensions
osapi_compute_extension = [nova.api.openstack.compute.contrib.standard_extensions]	(MultiStrOpt) osapi compute extension to load
osapi_compute_link_prefix = None	(StrOpt) Base URL that will be presented to users in links to the OpenStack Compute API
osapi_compute_listen = 0.0.0.0	(StrOpt) The IP address on which the OpenStack API will listen.
osapi_compute_listen_port = 8774	(IntOpt) The port on which the OpenStack API will listen.
osapi_compute_workers = None	(IntOpt) Number of workers for OpenStack API service. The default will be the number of CPUs available.
osapi_hide_server_address_states = building	(ListOpt) List of instance states that should hide network info
servicegroup_driver = db	(StrOpt) The driver for servicegroup service (valid options are: db, zk, mc)

Configuration option = Default value	Description
snapshot_name_template = snapshot-%s	(StrOpt) Template string to be used to generate snapshot names
use_forwarded_for = False	(BoolOpt) Treat X-Forwarded-For as the canonical remote address. Only enable this if you have a sanitizing proxy.

Table 2.12. Description of configuration options for apiv3

Configuration option = Default value	Description
[osapi_v3]	
enabled = False	(BoolOpt) Whether the V3 API is enabled or not
extensions_blacklist =	(ListOpt) A list of v3 API extensions to never load. Specify the extension aliases here.
extensions_whitelist =	(ListOpt) If the list is not empty then a v3 API extension will only be loaded if it exists in this list. Specify the extension aliases here.

Table 2.13. Description of configuration options for authentication

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = noauth	(StrOpt) The strategy to use for auth: noauth or keystone.

Table 2.14. Description of configuration options for auth_token

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) 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

Configuration option = Default value	Description
	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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. 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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

Table 2.15. Description of configuration options for availabilityzones

Configuration option = Default value	Description
[DEFAULT]	
default_availability_zone = nova	(StrOpt) Default compute node availability_zone
default_schedule_zone = None	(StrOpt) Availability zone to use when user doesn't specify one
internal_service_availability_zone = internal	(StrOpt) The availability_zone to show internal services under

Table 2.16. Description of configuration options for baremetal

Configuration option = Default value	Description
[baremetal]	
db_backend = sqlalchemy	(StrOpt) The backend to use for bare-metal database

Configuration option = Default value	Description
deploy_kernel = None	(StrOpt) Default kernel image ID used in deployment phase
deploy_ramdisk = None	(StrOpt) Default ramdisk image ID used in deployment phase
driver = nova.virt.baremetal.pxe.PXE	(StrOpt) Baremetal driver back-end (pxe or tilera)
flavor_extra_specs =	(ListOpt) A list of additional capabilities corresponding to flavor_extra_specs for this compute host to advertise. Valid entries are name=value, pairs For example, "key1:val1, key2:val2"
ipmi_power_retry = 10	(IntOpt) Maximal number of retries for IPMI operations
net_config_template = \$pybasedir/nova/virt/baremetal/net-dhcp.ubuntu.template	(StrOpt) Template file for injected network config
power_manager = nova.virt.baremetal.ipmi.IPMI	(StrOpt) Baremetal power management method
pxe_append_params = nofb nomodeset vga=normal	(StrOpt) Additional append parameters for baremetal PXE boot
pxe_bootfile_name = pxelinux.0	(StrOpt) This gets passed to Neutron as the bootfile dhcp parameter.
pxe_config_template = \$pybasedir/nova/virt/baremetal/pxe_config.template	(StrOpt) Template file for PXE configuration
pxe_deploy_timeout = 0	(IntOpt) Timeout for PXE deployments. Default: 0 (unlimited)
pxe_network_config = False	(BoolOpt) If set, pass the network configuration details to the initramfs via cmdline.
sql_connection = sqlite:/// \$state_path/baremetal_nova.sqlite	(StrOpt) The SQLAlchemy connection string used to connect to the bare-metal database
terminal = shellinaboxd	(StrOpt) Path to baremetal terminal program
terminal_cert_dir = None	(StrOpt) Path to baremetal terminal SSL cert(PEM)
terminal_pid_dir = \$state_path/baremetal/console	(StrOpt) Path to directory stores pidfiles of baremetal_terminal
tftp_root = /tftpboot	(StrOpt) Baremetal compute node's tftp root path
use_file_injection = False	(BoolOpt) If True, enable file injection for network info, files and admin password
use_unsafe_iscsi = False	(BoolOpt) Do not set this out of dev/test environments. If a node does not have a fixed PXE IP address, volumes are exported with globally opened ACL
vif_driver = nova.virt.baremetal.vif_driver.BareMetalVIFDriver	(StrOpt) Baremetal VIF driver.
virtual_power_host_key = None	(StrOpt) The ssh key for virtual power host_user
virtual_power_host_pass =	(StrOpt) Password for virtual power host_user
virtual_power_host_user =	(StrOpt) User to execute virtual power commands as
virtual_power_ssh_host =	(StrOpt) IP or name to virtual power host
virtual_power_ssh_port = 22	(IntOpt) Port to use for ssh to virtual power host
virtual_power_type = virsh	(StrOpt) Base command to use for virtual power(vbox, virsh)
[xenserver]	
vif_driver = nova.virt.xenapi.vif.XenAPIBridgeDriver	(StrOpt) The XenAPI VIF driver using XenServer Network APIs.

Table 2.17. Description of configuration options for ca

Configuration option = Default value	Description
[DEFAULT]	
ca_file = cacert.pem	(StrOpt) Filename of root CA
ca_path = \$state_path/CA	(StrOpt) Where we keep our root CA
cert_manager = nova.cert.manager.CertManager	(StrOpt) Full class name for the Manager for cert
cert_topic = cert	(StrOpt) The topic cert nodes listen on
crl_file = crl.pem	(StrOpt) Filename of root Certificate Revocation List
key_file = private/cakey.pem	(StrOpt) Filename of private key
keys_path = \$state_path/keys	(StrOpt) Where we keep our keys
project_cert_subject = /C=US/ST=California/O=OpenStack/OU=NovaDev/CN=project-ca-%.16s-%s	(StrOpt) Subject for certificate for projects, %s for project, timestamp
use_project_ca = False	(BoolOpt) Should we use a CA for each project?
user_cert_subject = /C=US/ST=California/O=OpenStack/OU=NovaDev/CN=%%.16s-%.16s-%s	(StrOpt) Subject for certificate for users, %s for project, user, timestamp
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients.
cert_file = None	(StrOpt) Certificate file to use when starting the server securely.
key_file = None	(StrOpt) Private key file to use when starting the server securely.

Table 2.18. Description of configuration options for cells

Configuration option = Default value	Description
[cells]	
call_timeout = 60	(IntOpt) Seconds to wait for response from a call to a cell.
capabilities = hypervisor=xenserver;kvm, os=linux;windows	(ListOpt) Key/Multi-value list with the capabilities of the cell
cell_type = compute	(StrOpt) Type of cell: api or compute
cells_config = None	(StrOpt) Configuration file from which to read cells configuration. If given, overrides reading cells from the database.
driver = nova.cells.rpc_driver.CellsRPCDriver	(StrOpt) Cells communication driver to use
enable = False	(BoolOpt) Enable cell functionality
instance_update_num_instances = 1	(IntOpt) Number of instances to update per periodic task run
instance_updated_at_threshold = 3600	(IntOpt) Number of seconds after an instance was updated or deleted to continue to update cells
manager = nova.cells.manager.CellsManager	(StrOpt) Manager for cells
max_hop_count = 10	(IntOpt) Maximum number of hops for cells routing.
mute_child_interval = 300	(IntOpt) Number of seconds after which a lack of capability and capacity updates signals the child cell is to be treated as a mute.
mute_weight_multiplier = -10.0	(FloatOpt) Multiplier used to weigh mute children. (The value should be negative.)
mute_weight_value = 1000.0	(FloatOpt) Weight value assigned to mute children. (The value should be positive.)
name = nova	(StrOpt) Name of this cell

Configuration option = Default value	Description
offset_weight_multiplier = 1.0	(FloatOpt) Multiplier used to weigh offset weigher.
reserve_percent = 10.0	(FloatOpt) Percentage of cell capacity to hold in reserve. Affects both memory and disk utilization
topic = cells	(StrOpt) The topic cells nodes listen on

Table 2.19. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
bindir = /usr/local/bin	(StrOpt) Directory where nova binaries are installed
compute_topic = compute	(StrOpt) The topic compute nodes listen on
console_topic = console	(StrOpt) The topic console proxy nodes listen on
consoleauth_topic = consoleauth	(StrOpt) The topic console auth proxy nodes listen on
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
host = oslo	(StrOpt) 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
lock_path = None	(StrOpt) Directory to use for lock files.
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
my_ip = 10.0.0.1	(StrOpt) IP address of this host
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = notifications	(ListOpt) AMQP topic used for OpenStack notifications.
notify_api_faults = False	(BoolOpt) If set, send api.fault notifications on caught exceptions in the API service.
notify_on_state_change = None	(StrOpt) If set, send compute.instance.update notifications on instance state changes. Valid values are None for no notifications, "vm_state" for notifications on VM state changes, or "vm_and_task_state" for notifications on VM and task state changes.
port = 6379	(IntOpt) Use this port to connect to redis host.
pybasedir = /usr/lib/python/site-packages	(StrOpt) Directory where the nova python module is installed
report_interval = 10	(IntOpt) Seconds between nodes reporting state to datastore
rootwrap_config = /etc/nova/rootwrap.conf	(StrOpt) Path to the rootwrap configuration file to use for running commands as root
service_down_time = 60	(IntOpt) Maximum time since last check-in for up service
state_path = \$pybasedir	(StrOpt) Top-level directory for maintaining nova's state
tempdir = None	(StrOpt) Explicitly specify the temporary working directory
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.

Table 2.20. Description of configuration options for compute

Configuration option = Default value	Description
[DEFAULT]	
compute_available_monitors = ['nova.compute.monitors.all_monitors']	(MultiStrOpt) Monitor classes available to the compute which may be specified more than once.

Configuration option = Default value	Description
compute_driver = None	(StrOpt) Driver to use for controlling virtualization. Options include: libvirt.LibvirtDriver, xenapi.XenAPIDriver, fake.FakeDriver, baremetal.BareMetalDriver, vmwareapi.VMwareESXDriver, vmwareapi.VMwareVCDriver
compute_manager = nova.compute.manager.ComputeManager	(StrOpt) Full class name for the Manager for compute
compute_monitors =	(ListOpt) A list of monitors that can be used for getting compute metrics.
compute_stats_class = nova.compute.stats.Stats	(StrOpt) Class that will manage stats for the local compute host
console_host = oslo	(StrOpt) Console proxy host to use to connect to instances on this host.
console_manager = nova.console.manager.ConsoleProxyManager	(StrOpt) Full class name for the Manager for console proxy
default_flavor = m1.small	(StrOpt) Default flavor to use for the EC2 API only. The Nova API does not support a default flavor.
default_notification_level = INFO	(StrOpt) Default notification level for outgoing notifications
default_publisher_id = None	(StrOpt) Default publisher_id for outgoing notifications
enable_instance_password = True	(BoolOpt) Enables returning of the instance password by the relevant server API calls such as create, rebuild or rescue. If the hypervisor does not support password injection then the password returned will not be correct
heal_instance_info_cache_interval = 60	(IntOpt) Number of seconds between instance info_cache self healing updates
image_cache_manager_interval = 2400	(IntOpt) Number of seconds to wait between runs of the image cache manager
image_cache_subdirectory_name = _base	(StrOpt) Where cached images are stored under \$instances_path. This is NOT the full path - just a folder name. For per-compute-host cached images, set to _base_\$my_ip
image_handlers = download	(ListOpt) Specifies which image handler extension names to use for handling images. The first extension in the list which can handle the image with a suitable location will be used.
instance_build_timeout = 0	(IntOpt) Amount of time in seconds an instance can be in BUILD before going into ERROR status. Set to 0 to disable.
instance_delete_interval = 300	(IntOpt) Interval in seconds for retrying failed instance file deletes
instance_usage_audit = False	(BoolOpt) Generate periodic compute.instance.exists notifications
instance_usage_audit_period = month	(StrOpt) Time period to generate instance usages for. Time period must be hour, day, month or year
instances_path = \$state_path/instances	(StrOpt) Where instances are stored on disk
maximum_instance_delete_attempts = 5	(IntOpt) The number of times to attempt to reap an instance's files.
reboot_timeout = 0	(IntOpt) Automatically hard reboot an instance if it has been stuck in a rebooting state longer than N seconds. Set to 0 to disable.
reclaim_instance_interval = 0	(IntOpt) Interval in seconds for reclaiming deleted instances
resize_confirm_window = 0	(IntOpt) Automatically confirm resizes after N seconds. Set to 0 to disable.

Configuration option = Default value	Description
resume_guests_state_on_host_boot = False	(BoolOpt) Whether to start guests that were running before the host rebooted
running_deleted_instance_action = reap	(StrOpt) Action to take if a running deleted instance is detected. Valid options are 'noop', 'log', 'shutdown', or 'reap'. Set to 'noop' to take no action.
running_deleted_instance_poll_interval = 1800	(IntOpt) Number of seconds to wait between runs of the cleanup task.
running_deleted_instance_timeout = 0	(IntOpt) Number of seconds after being deleted when a running instance should be considered eligible for cleanup.
shelved_offload_time = 0	(IntOpt) Time in seconds before a shelved instance is eligible for removing from a host. -1 never offload, 0 offload when shelved
shelved_poll_interval = 3600	(IntOpt) Interval in seconds for polling shelved instances to offload
sync_power_state_interval = 600	(IntOpt) Interval to sync power states between the database and the hypervisor
vif_plugging_is_fatal = True	(BoolOpt) Fail instance boot if vif plugging fails
vif_plugging_timeout = 300	(IntOpt) Number of seconds to wait for neutron vif plugging events to arrive before continuing or failing (see vif_plugging_is_fatal). If this is set to zero and vif_plugging_is_fatal is False, events should not be expected to arrive at all.

Table 2.21. Description of configuration options for conductor

Configuration option = Default value	Description
[DEFAULT]	
migrate_max_retries = -1	(IntOpt) Number of times to retry live-migration before failing. If == -1, try until out of hosts. If == 0, only try once, no retries.
[conductor]	
manager = nova.conductor.manager.ConductorManager	(StrOpt) Full class name for the Manager for conductor
topic = conductor	(StrOpt) The topic on which conductor nodes listen
use_local = False	(BoolOpt) Perform nova-conductor operations locally
workers = None	(IntOpt) Number of workers for OpenStack Conductor service. The default will be the number of CPUs available.

Table 2.22. Description of configuration options for configdrive

Configuration option = Default value	Description
[DEFAULT]	
config_drive_format = iso9660	(StrOpt) Config drive format. One of iso9660 (default) or vfat
config_drive_skip_versions = 1.0 2007-01-19 2007-03-01 2007-08-29 2007-10-10 2007-12-15 2008-02-01 2008-09-01	(StrOpt) List of metadata versions to skip placing into the config drive
config_drive_tmpdir = None	(StrOpt) Where to put temporary files associated with config drive creation
force_config_drive = None	(StrOpt) Set to force injection to take place on a config drive (if set, valid options are: always)
mkisofs_cmd = genisoimage	(StrOpt) Name and optionally path of the tool used for ISO image creation
[hyperv]	

Configuration option = Default value	Description
config_drive_cdrom = False	(BoolOpt) Attaches the Config Drive image as a cdrom drive instead of a disk drive
config_drive_inject_password = False	(BoolOpt) Sets the admin password in the config drive image

Table 2.23. Description of configuration options for console

Configuration option = Default value	Description
[DEFAULT]	
console_public_hostname = oslo	(StrOpt) Publicly visible name for this console host
console_token_ttl = 600	(IntOpt) How many seconds before deleting tokens
consoleauth_manager = nova.consoleauth.manager.ConsoleAuthManager	(StrOpt) Manager for console auth

Table 2.24. Description of configuration options for db

Configuration option = Default value	Description
[DEFAULT]	
db_driver = nova.db	(StrOpt) The driver to use for database access
[cells]	
db_check_interval = 60	(IntOpt) Interval, in seconds, for getting fresh cell information from the database.
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
connection = None	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
db_retry_interval = 1	(IntOpt) seconds between db connection retries
idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
mysql_sql_mode = None	(StrOpt) The SQL mode to be used for MySQL sessions (default is empty, meaning do not override any server-side SQL mode setting)
pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy

Configuration option = Default value	Description
retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
slave_connection = None	(StrOpt) The SQLAlchemy connection string used to connect to the slave database
sqlite_db = nova.sqlite	(StrOpt) The file name to use with SQLite
sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost

Table 2.25. Description of configuration options for ec2

Configuration option = Default value	Description
[DEFAULT]	
ec2_dmz_host = \$my_ip	(StrOpt) The internal IP address of the EC2 API server
ec2_host = \$my_ip	(StrOpt) The IP address of the EC2 API server
ec2_listen = 0.0.0.0	(StrOpt) The IP address on which the EC2 API will listen.
ec2_listen_port = 8773	(IntOpt) The port on which the EC2 API will listen.
ec2_path = /services/Cloud	(StrOpt) The path prefix used to call the ec2 API server
ec2_port = 8773	(IntOpt) The port of the EC2 API server
ec2_private_dns_show_ip = False	(BoolOpt) Return the IP address as private dns hostname in describe instances
ec2_scheme = http	(StrOpt) The protocol to use when connecting to the EC2 API server (http, https)
ec2_strict_validation = True	(BoolOpt) Validate security group names according to EC2 specification
ec2_timestamp_expiry = 300	(IntOpt) Time in seconds before ec2 timestamp expires
ec2_workers = None	(IntOpt) Number of workers for EC2 API service. The default will be equal to the number of CPUs available.
keystone_ec2_url = http://localhost:5000/v2.0/ec2tokens	(StrOpt) URL to get token from ec2 request.
lockout_attempts = 5	(IntOpt) Number of failed auths before lockout.
lockout_minutes = 15	(IntOpt) Number of minutes to lockout if triggered.
lockout_window = 15	(IntOpt) Number of minutes for lockout window.
region_list =	(ListOpt) List of region=fqdn pairs separated by commas

Table 2.26. Description of configuration options for fping

Configuration option = Default value	Description
[DEFAULT]	
fping_path = /usr/sbin/fping	(StrOpt) Full path to fping.

Table 2.27. Description of configuration options for glance

Configuration option = Default value	Description
[DEFAULT]	
allowed_direct_url_schemes =	(ListOpt) A list of url scheme that can be downloaded directly via the direct_url. Currently supported schemes: [file].
glance_api_insecure = False	(BoolOpt) Allow to perform insecure SSL (https) requests to glance

Configuration option = Default value	Description
glance_api_servers = \$glance_host:\$glance_port	(ListOpt) A list of the glance api servers available to nova. Prefix with https:// for ssl-based glance api servers. ([hostname ip]:port)
glance_host = \$my_ip	(StrOpt) Default glance hostname or IP address
glance_num_retries = 0	(IntOpt) Number of retries when downloading an image from glance
glance_port = 9292	(IntOpt) Default glance port
glance_protocol = http	(StrOpt) Default protocol to use when connecting to glance. Set to https for SSL.
osapi_glance_link_prefix = None	(StrOpt) Base URL that will be presented to users in links to glance resources
[image_file_url]	
filesystems =	(ListOpt) List of file systems that are configured in this file in the image_file_url:<list entry name> sections

Table 2.28. Description of configuration options for hyperv

Configuration option = Default value	Description
[hyperv]	
dynamic_memory_ratio = 1.0	(FloatOpt) Enables dynamic memory allocation (ballooning) when set to a value greater than 1. The value expresses the ratio between the total RAM assigned to an instance and its startup RAM amount. For example a ratio of 2.0 for an instance with 1024MB of RAM implies 512MB of RAM allocated at startup
enable_instance_metrics_collection = False	(BoolOpt) Enables metrics collections for an instance by using Hyper-V's metric APIs. Collected data can be retrieved by other apps and services, e.g.: Ceilometer. Requires Hyper-V / Windows Server 2012 and above
force_hyperv_utils_v1 = False	(BoolOpt) Force V1 WMI utility classes
instances_path_share =	(StrOpt) The name of a Windows share name mapped to the "instances_path" dir and used by the resize feature to copy files to the target host. If left blank, an administrative share will be used, looking for the same "instances_path" used locally
limit_cpu_features = False	(BoolOpt) Required for live migration among hosts with different CPU features
mounted_disk_query_retry_count = 10	(IntOpt) The number of times to retry checking for a disk mounted via iSCSI.
mounted_disk_query_retry_interval = 5	(IntOpt) Interval between checks for a mounted iSCSI disk, in seconds.
qemu_img_cmd = qemu-img.exe	(StrOpt) Path of qemu-img command which is used to convert between different image types
vswitch_name = None	(StrOpt) External virtual switch Name, if not provided, the first external virtual switch is used

Table 2.29. Description of configuration options for hypervisor

Configuration option = Default value	Description
[DEFAULT]	
default_ephemeral_format = None	(StrOpt) The default format an ephemeral_volume will be formatted with on creation.
force_raw_images = True	(BoolOpt) Force backing images to raw format

Configuration option = Default value	Description
preallocate_images = none	(StrOpt) VM image preallocation mode: "none" => no storage provisioning is done up front, "space" => storage is fully allocated at instance start
rescue_timeout = 0	(IntOpt) Automatically unrescue an instance after N seconds. Set to 0 to disable.
timeout_nbd = 10	(IntOpt) Amount of time, in seconds, to wait for NBD device start up.
use_cow_images = True	(BoolOpt) Whether to use cow images
vcpu_pin_set = None	(StrOpt) Defines which pcpus that instance vcpus can use. For example, "4-12,^8,15"
virt_mkfs = []	(MultiStrOpt) Name of the mkfs commands for ephemeral device. The format is <os_type>=<mkfs command>
[libvirt]	
block_migration_flag = VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_NON_SHARED_INC	(StrOpt) Migration flags to be set for block migration
disk_cachemodes =	(ListOpt) Specific cachemodes to use for different disk types e.g: file=directsync,block=none
images_rbd_ceph_conf =	(StrOpt) Path to the ceph configuration file to use
images_rbd_pool = rbd	(StrOpt) The RADOS pool in which rbd volumes are stored
images_type = default	(StrOpt) VM Images format. Acceptable values are: raw, qcow2, lvm, rbd, default. If default is specified, then use_cow_images flag is used instead of this one.
images_volume_group = None	(StrOpt) LVM Volume Group that is used for VM images, when you specify images_type=lvm.
inject_key = False	(BoolOpt) Inject the ssh public key at boot time
inject_partition = -2	(IntOpt) The partition to inject to : -2 => disable, -1 => inspect (libguestfs only), 0 => not partitioned, >0 => partition number
inject_password = False	(BoolOpt) Inject the admin password at boot time, without an agent.
iscsi_use_multipath = False	(BoolOpt) Use multipath connection of the iSCSI volume
iser_use_multipath = False	(BoolOpt) Use multipath connection of the iSER volume
rescue_image_id = None	(StrOpt) Rescue ami image
rescue_kernel_id = None	(StrOpt) Rescue aki image
rescue_ramdisk_id = None	(StrOpt) Rescue ari image
snapshot_compression = False	(BoolOpt) Compress snapshot images when possible. This currently applies exclusively to qcow2 images
snapshot_image_format = None	(StrOpt) Snapshot image format (valid options are : raw, qcow2, vmdk, vdi). Defaults to same as source image
sparse_logical_volumes = False	(BoolOpt) Create sparse logical volumes (with virtualsize) if this flag is set to True.
use_usb_tablet = True	(BoolOpt) Sync virtual and real mouse cursors in Windows VMs
use_virtio_for_bridges = True	(BoolOpt) Use virtio for bridge interfaces with KVM/ QEMU

Table 2.30. Description of configuration options for ipv6

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
fixed_range_v6 = fd00::/48	(StrOpt) Fixed IPv6 address block
gateway_v6 = None	(StrOpt) Default IPv6 gateway
ipv6_backend = rfc2462	(StrOpt) Backend to use for IPv6 generation
use_ipv6 = False	(BoolOpt) Use IPv6

Table 2.31. Description of configuration options for keymgr

Configuration option = Default value	Description
[keymgr]	
api_class = nova.keymgr.conf_key_mgr.ConfKeyManager	(StrOpt) The full class name of the key manager API class
fixed_key = None	(StrOpt) Fixed key returned by key manager, specified in hex

Table 2.32. Description of configuration options for ldap

Configuration option = Default value	Description
[DEFAULT]	
ldap_dns_base_dn = ou=hosts,dc=example,dc=org	(StrOpt) Base DN for DNS entries in LDAP
ldap_dns_password = password	(StrOpt) Password for LDAP DNS
ldap_dns_servers = ['dns.example.org']	(MultiStrOpt) DNS Servers for LDAP DNS driver
ldap_dns_soa_expiry = 86400	(StrOpt) Expiry interval (in seconds) for LDAP DNS driver Statement of Authority
ldap_dns_soa_hostmaster = hostmaster@example.org	(StrOpt) Hostmaster for LDAP DNS driver Statement of Authority
ldap_dns_soa_minimum = 7200	(StrOpt) Minimum interval (in seconds) for LDAP DNS driver Statement of Authority
ldap_dns_soa_refresh = 1800	(StrOpt) Refresh interval (in seconds) for LDAP DNS driver Statement of Authority
ldap_dns_soa_retry = 3600	(StrOpt) Retry interval (in seconds) for LDAP DNS driver Statement of Authority
ldap_dns_url = ldap://ldap.example.com:389	(StrOpt) URL for LDAP server which will store DNS entries
ldap_dns_user = uid=admin,ou=people,dc=example,dc=org	(StrOpt) User for LDAP DNS

Table 2.33. Description of configuration options for libvirt

Configuration option = Default value	Description
[DEFAULT]	
remove_unused_base_images = True	(BoolOpt) Should unused base images be removed?
remove_unused_original_minimum_age_seconds = 86400	(IntOpt) Unused unresized base images younger than this will not be removed
[libvirt]	
checksum_base_images = False	(BoolOpt) Write a checksum for files in _base to disk
checksum_interval_seconds = 3600	(IntOpt) How frequently to checksum base images
connection_uri =	(StrOpt) Override the default libvirt URI (which is dependent on virt_type)
cpu_mode = None	(StrOpt) Set to "host-model" to clone the host CPU feature flags; to "host-passthrough" to use the host CPU model exactly; to "custom" to use a named CPU model; to "none" to not set any CPU model. If virt_type="kvm qemu", it will default to "host-model", otherwise it will default to "none"

Configuration option = Default value	Description
cpu_model = None	(StrOpt) Set to a named libvirt CPU model (see names listed in /usr/share/libvirt/cpu_map.xml). Only has effect if cpu_mode="custom" and virt_type="kvm qemu"
disk_prefix = None	(StrOpt) Override the default disk prefix for the devices attached to a server, which is dependent on virt_type. (valid options are: sd, xvd, uvd, vd)
image_info_filename_pattern = \$instances_path/\$image_cache_subdirectory_name/(image)s.info	(StrOpt) Allows image information files to be stored in non-standard locations
remove_unused_kernels = False	(BoolOpt) Should unused kernel images be removed? This is only safe to enable if all compute nodes have been updated to support this option. This will be enabled by default in future.
remove_unused_resized_minimum_age_seconds = 3600	(IntOpt) Unused resized base images younger than this will not be removed
rng_dev_path = None	(StrOpt) A path to a device that will be used as source of entropy on the host. Permitted options are: /dev/random or /dev/hwrng
snapshots_directory = \$instances_path/snapshots	(StrOpt) Location where libvirt driver will store snapshots before uploading them to image service
vif_driver = nova.virt.libvirt.vif.LibvirtGenericVIFDriver	(StrOpt) DEPRECATED. The libvirt VIF driver to configure the VIFs. This option is deprecated and will be removed in the Juno release.
virt_type = kvm	(StrOpt) Libvirt domain type (valid options are: kvm, lxc, qemu, uml, xen)
volume_clear = zero	(StrOpt) Method used to wipe old volumes (valid options are: none, zero, shred)
volume_clear_size = 0	(IntOpt) Size in MiB to wipe at start of old volumes. 0 => all
volume_drivers = iscsi=nova.virt.libvirt.volume.LibvirtISCSIVolumeDriver, iser=nova.virt.libvirt.volume.LibvirtISERVolumeDriver, local=nova.virt.libvirt.volume.LibvirtVolumeDriver, fake=nova.virt.libvirt.volume.LibvirtFakeVolumeDriver, rbd=nova.virt.libvirt.volume.LibvirtNetVolumeDriver, sheepdog=nova.virt.libvirt.volume.LibvirtNetVolumeDriver, nfs=nova.virt.libvirt.volume.LibvirtNFSVolumeDriver, aoe=nova.virt.libvirt.volume.LibvirtAOEVolumeDriver, glusterfs=nova.virt.libvirt.volume.LibvirtGlusterfsVolumeDriver, fibre_channel=nova.virt.libvirt.volume.LibvirtFibreChannelVolumeDriver, scalacity=nova.virt.libvirt.volume.LibvirtScalacityVolumeDriver	(ListOpt) Libvirt handlers for remote volumes.
wait_soft_reboot_seconds = 120	(IntOpt) Number of seconds to wait for instance to shut down after soft reboot request is made. We fall back to hard reboot if instance does not shutdown within this window.

Table 2.34. Description of configuration options for livemigration

Configuration option = Default value	Description
[DEFAULT]	
live_migration_retry_count = 30	(IntOpt) Number of 1 second retries needed in live_migration
[libvirt]	
live_migration_bandwidth = 0	(IntOpt) Maximum bandwidth to be used during migration, in Mbps
live_migration_flag = VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER	(StrOpt) Migration flags to be set for live migration

Configuration option = Default value	Description
live_migration_uri = qemu+tcp://%/s/system	(StrOpt) Migration target URI (any included "%s" is replaced with the migration target hostname)

Table 2.35. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
default_log_levels = amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN	(ListOpt) list of logger=LEVEL pairs
fatal_deprecations = False	(BoolOpt) make deprecations fatal
fatal_exception_format_errors = False	(BoolOpt) Make exception message format errors fatal
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user)s %(tenant)s] %(instance)s %(message)s	(StrOpt) format string to use for log messages with context
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) prefix each line of exception output with this format
publish_errors = False	(BoolOpt) publish error events
syslog_log_facility = LOG_USER	(StrOpt) syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and then will be changed in J to honor RFC5424
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format

Configuration option = Default value	Description
	without APP-NAME is deprecated in I, and will be removed in J.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 2.36. Description of configuration options for metadata

Configuration option = Default value	Description
[DEFAULT]	
metadata_host = \$my_ip	(StrOpt) The IP address for the metadata API server
metadata_listen = 0.0.0.0	(StrOpt) The IP address on which the metadata API will listen.
metadata_listen_port = 8775	(IntOpt) The port on which the metadata API will listen.
metadata_manager = nova.api.manager.MetadataManager	(StrOpt) OpenStack metadata service manager
metadata_port = 8775	(IntOpt) The port for the metadata API port
metadata_workers = None	(IntOpt) Number of workers for metadata service. The default will be the number of CPUs available.
vendordata_driver = nova.api.metadata.vendordata_json.JsonFileVendorData	(StrOpt) Driver to use for vendor data
vendordata_jsonfile_path = None	(StrOpt) File to load json formatted vendor data from

Table 2.37. Description of configuration options for network

Configuration option = Default value	Description
[DEFAULT]	
allow_same_net_traffic = True	(BoolOpt) Whether to allow network traffic from same network
auto_assign_floating_ip = False	(BoolOpt) Autoassigning floating IP to VM
cnt_vpn_clients = 0	(IntOpt) Number of addresses reserved for vpn clients
create_unique_mac_address_attempts = 5	(IntOpt) Number of attempts to create unique mac address
default_access_ip_network_name = None	(StrOpt) Name of network to use to set access IPs for instances
default_floating_pool = nova	(StrOpt) Default pool for floating IPs
defer iptables apply = False	(BoolOpt) Whether to batch up the application of IPTables rules during a host restart and apply all at the end of the init phase
dhcp_domain = novalocal	(StrOpt) Domain to use for building the hostnames
dhcp_lease_time = 120	(IntOpt) Lifetime of a DHCP lease in seconds
dhcpbridge = \$bindir/nova-dhcpbridge	(StrOpt) Location of nova-dhcpbridge
dhcpbridge_flagfile = ["/etc/nova/nova-dhcpbridge.conf"]	(MultiStrOpt) Location of flagfiles for dhcpbridge
dns_server = []	(MultiStrOpt) If set, uses specific DNS server for dnsmasq. Can be specified multiple times.
dns_update_periodic_interval = -1	(IntOpt) Number of seconds to wait between runs of updates to DNS entries.
dnsmasq_config_file =	(StrOpt) Override the default dnsmasq settings with this file
firewall_driver = None	(StrOpt) Firewall driver (defaults to hypervisor specific iptables driver)
fixed_ip_disassociate_timeout = 600	(IntOpt) Seconds after which a deallocated IP is disassociated

Configuration option = Default value	Description
flat_injected = False	(BoolOpt) Whether to attempt to inject network setup into guest
flat_interface = None	(StrOpt) FlatDhcp will bridge into this interface if set
flat_network_bridge = None	(StrOpt) Bridge for simple network instances
flat_network_dns = 8.8.4.4	(StrOpt) DNS server for simple network
floating_ip_dns_manager = nova.network.noop_dns_driver.NoopDNSDriver	(StrOpt) Full class name for the DNS Manager for floating IPs
force_dhcp_release = True	(BoolOpt) If True, send a dhcp release on instance termination
force_snat_range = []	(MultiStrOpt) Traffic to this range will always be snatted to the fallback ip, even if it would normally be bridged out of the node. Can be specified multiple times.
forward_bridge_interface = ['all']	(MultiStrOpt) An interface that bridges can forward to. If this is set to all then all traffic will be forwarded. Can be specified multiple times.
gateway = None	(StrOpt) Default IPv4 gateway
injected_network_template = \$pybasedir/nova/virt/interfaces.template	(StrOpt) Template file for injected network
instance_dns_domain =	(StrOpt) Full class name for the DNS Zone for instance IPs
instance_dns_manager = nova.network.noop_dns_driver.NoopDNSDriver	(StrOpt) Full class name for the DNS Manager for instance IPs
iptables_bottom_regex =	(StrOpt) Regular expression to match iptables rule that should always be on the bottom.
iptables_drop_action = DROP	(StrOpt) The table that iptables to jump to when a packet is to be dropped.
iptables_top_regex =	(StrOpt) Regular expression to match iptables rule that should always be on the top.
l3_lib = nova.network.l3.LinuxNetL3	(StrOpt) Indicates underlying L3 management library
linuxnet_interface_driver = nova.network.linux_net.LinuxBridgeInterfaceDriver	(StrOpt) Driver used to create ethernet devices.
linuxnet_ovs_integration_bridge = br-int	(StrOpt) Name of Open vSwitch bridge used with linuxnet
multi_host = False	(BoolOpt) Default value for multi_host in networks. Also, if set, some rpc network calls will be sent directly to host.
network_allocate_retries = 0	(IntOpt) Number of times to retry network allocation on failures
network_api_class = nova.network.api.API	(StrOpt) The full class name of the network API class to use
network_device_mtu = None	(IntOpt) MTU setting for network interface
network_driver = nova.network.linux_net	(StrOpt) Driver to use for network creation
network_manager = nova.network.manager.VlanManager	(StrOpt) Full class name for the Manager for network
network_size = 256	(IntOpt) Number of addresses in each private subnet
network_topic = network	(StrOpt) The topic network nodes listen on
networks_path = \$state_path/networks	(StrOpt) Location to keep network config files
num_networks = 1	(IntOpt) Number of networks to support
ovs_vsctl_timeout = 120	(IntOpt) Amount of time, in seconds, that ovs_vsctl should wait for a response from the database. 0 is to wait forever.
public_interface = eth0	(StrOpt) Interface for public IP addresses
routing_source_ip = \$my_ip	(StrOpt) Public IP of network host
security_group_api = nova	(StrOpt) The full class name of the security API class

Configuration option = Default value	Description
send_arp_for_ha = False	(BoolOpt) Send gratuitous ARPs for HA setup
send_arp_for_ha_count = 3	(IntOpt) Send this many gratuitous ARPs for HA setup
share_dhcp_address = False	(BoolOpt) If True in multi_host mode, all compute hosts share the same dhcp address. The same IP address used for DHCP will be added on each nova-network node which is only visible to the vms on the same host.
teardown_unused_network_gateway = False	(BoolOpt) If True, unused gateway devices (VLAN and bridge) are deleted in VLAN network mode with multi hosted networks
update_dns_entries = False	(BoolOpt) If True, when a DNS entry must be updated, it sends a fanout cast to all network hosts to update their DNS entries in multi host mode
use_network_dns_servers = False	(BoolOpt) If set, uses the dns1 and dns2 from the network ref. as dns servers.
use_neutron_default_nets = False	(StrOpt) Control for checking for default networks
use_single_default_gateway = False	(BoolOpt) Use single default gateway. Only first nic of vm will get default gateway from dhcp server
vlan_interface = None	(StrOpt) VLANs will bridge into this interface if set
vlan_start = 100	(IntOpt) First VLAN for private networks
[vmware]	
vlan_interface = vmnic0	(StrOpt) Physical ethernet adapter name for vlan networking

Table 2.38. Description of configuration options for neutron

Configuration option = Default value	Description
[DEFAULT]	
neutron_admin_auth_url = http://localhost:5000/v2.0	(StrOpt) Authorization URL for connecting to neutron in admin context
neutron_admin_password = None	(StrOpt) Password for connecting to neutron in admin context
neutron_admin_tenant_id = None	(StrOpt) Tenant id for connecting to neutron in admin context
neutron_admin_tenant_name = None	(StrOpt) Tenant name for connecting to neutron in admin context. This option is mutually exclusive with neutron_admin_tenant_id. Note that with Keystone V3 tenant names are only unique within a domain.
neutron_admin_username = None	(StrOpt) Username for connecting to neutron in admin context
neutron_api_insecure = False	(BoolOpt) If set, ignore any SSL validation issues
neutron_auth_strategy = keystone	(StrOpt) Authorization strategy for connecting to neutron in admin context
neutron_ca_certificates_file = None	(StrOpt) Location of CA certificates file to use for neutron client requests.
neutron_default_tenant_id = default	(StrOpt) Default tenant id when creating neutron networks
neutron_extension_sync_interval = 600	(IntOpt) Number of seconds before querying neutron for extensions
neutron_metadata_proxy_shared_secret =	(StrOpt) Shared secret to validate proxies Neutron metadata requests
neutron_ovs_bridge = br-int	(StrOpt) Name of Integration Bridge used by Open vSwitch

Configuration option = Default value	Description
neutron_region_name = None	(StrOpt) Region name for connecting to neutron in admin context
neutron_url = http://127.0.0.1:9696	(StrOpt) URL for connecting to neutron
neutron_url_timeout = 30	(IntOpt) Timeout value for connecting to neutron in seconds
service_neutron_metadata_proxy = False	(BoolOpt) Set flag to indicate Neutron will proxy metadata requests and resolve instance ids.

Table 2.39. Description of configuration options for pci

Configuration option = Default value	Description
[DEFAULT]	
pci_alias = []	(MultiStrOpt) An alias for a PCI passthrough device requirement. This allows users to specify the alias in the extra_spec for a flavor, without needing to repeat all the PCI property requirements. For example: pci_alias = { "name": "QuicAssist", "product_id": "0443", "vendor_id": "8086", "device_type": "ACCEL" } defines an alias for the Intel QuickAssist card. (multi valued)
pci_passthrough_whitelist = []	(MultiStrOpt) White list of PCI devices available to VMs. For example: pci_passthrough_whitelist = [{"vendor_id": "8086", "product_id": "0443"}]

Table 2.40. Description of configuration options for periodic

Configuration option = Default value	Description
[DEFAULT]	
periodic_enable = True	(BoolOpt) Enable periodic tasks
periodic_fuzzy_delay = 60	(IntOpt) Range of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
run_external_periodic_tasks = True	(BoolOpt) Some periodic tasks can be run in a separate process. Should we run them here?

Table 2.41. Description of configuration options for policy

Configuration option = Default value	Description
[DEFAULT]	
allow_instance_snapshots = True	(BoolOpt) Permit instance snapshot operations.
allow_migrate_to_same_host = False	(BoolOpt) Allow migrate machine to the same host. Useful when testing in single-host environments.
allow_resize_to_same_host = False	(BoolOpt) Allow destination machine to match source for resize. Useful when testing in single-host environments.
max_age = 0	(IntOpt) Number of seconds between subsequent usage refreshes
max_local_block_devices = 3	(IntOpt) Maximum number of devices that will result in a local image being created on the hypervisor node. Setting this to 0 means nova will allow only boot from volume. A negative number means unlimited.
osapi_compute_unique_server_name_scope =	(StrOpt) When set, compute API will consider duplicate hostnames invalid within the specified scope, regardless of case. Should be empty, "project" or "global".
osapi_max_limit = 1000	(IntOpt) The maximum number of items returned in a single response from a collection resource

Configuration option = Default value	Description
osapi_max_request_body_size = 114688	(IntOpt) The maximum body size per each osapi request(bytes)
password_length = 12	(IntOpt) Length of generated instance admin passwords
policy_default_rule = default	(StrOpt) Rule checked when requested rule is not found
policy_file = policy.json	(StrOpt) JSON file representing policy
reservation_expire = 86400	(IntOpt) Number of seconds until a reservation expires
resize_fs_using_block_device = False	(BoolOpt) Attempt to resize the filesystem by accessing the image over a block device. This is done by the host and may not be necessary if the image contains a recent version of cloud-init. Possible mechanisms require the nbd driver (for qcow and raw), or loop (for raw).
until_refresh = 0	(IntOpt) Count of reservations until usage is refreshed

Table 2.42. Description of configuration options for quota

Configuration option = Default value	Description
[DEFAULT]	
bandwidth_poll_interval = 600	(IntOpt) Interval to pull network bandwidth usage info. Not supported on all hypervisors. Set to 0 to disable.
enable_network_quota = False	(BoolOpt) Enables or disables quota checking for tenant networks
quota_cores = 20	(IntOpt) Number of instance cores allowed per project
quota_driver = nova.quota.DbQuotaDriver	(StrOpt) Default driver to use for quota checks
quota_fixed_ips = -1	(IntOpt) Number of fixed IPs allowed per project (this should be at least the number of instances allowed)
quota_floating_ips = 10	(IntOpt) Number of floating IPs allowed per project
quota_injected_file_content_bytes = 10240	(IntOpt) Number of bytes allowed per injected file
quota_injected_file_path_bytes = 255	(IntOpt) Number of bytes allowed per injected file path
quota_injected_files = 5	(IntOpt) Number of injected files allowed
quota_instances = 10	(IntOpt) Number of instances allowed per project
quota_key_pairs = 100	(IntOpt) Number of key pairs per user
quota_metadata_items = 128	(IntOpt) Number of metadata items allowed per instance
quota_ram = 51200	(IntOpt) Megabytes of instance RAM allowed per project
quota_security_group_rules = 20	(IntOpt) Number of security rules per security group
quota_security_groups = 10	(IntOpt) Number of security groups per project
[cells]	
bandwidth_update_interval = 600	(IntOpt) Seconds between bandwidth updates for cells.

Table 2.43. Description of configuration options for rdp

Configuration option = Default value	Description
[rdp]	
enabled = False	(BoolOpt) Enable RDP related features
html5_proxy_base_url = http://127.0.0.1:6083/	(StrOpt) Location of RDP html5 console proxy, in the form "http://127.0.0.1:6083/"

Table 2.44. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
password = None	(StrOpt) Password for Redis server (optional).

Table 2.45. Description of configuration options for rootwrap

Configuration option = Default value	Description
[DEFAULT]	
filters_path = /etc/nova/rootwrap.d,/usr/share/nova/rootwrap	List of directories to load filter definitions from (separated by ','). These directories MUST all be only writeable by root !
exec_dirs = /sbin,/usr/sbin,/bin,/usr/bin	List of directories to search executables in, in case filters do not explicitly specify a full path (separated by ',') If not specified, defaults to system PATH environment variable. These directories MUST all be only writeable by root !
use_syslog = False	Enable logging to syslog Default value is False
syslog_log_facility = syslog	Which syslog facility to use. Valid values include auth, authpriv, syslog, user0, user1... Default value is 'syslog'
syslog_log_level = ERROR	Which messages to log. INFO means log all usage ERROR means only log unsuccessful attempts

Table 2.46. Description of configuration options for rpc_all

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
control_exchange = openstack	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
qpid_password =	(StrOpt) Password for Qpid connection.
qpid_port = 5672	(IntOpt) Qpid broker port.
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker

Configuration option = Default value	Description
	federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for Qpid connection.
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
rabbit_password = guest	(StrOpt) The RabbitMQ password.
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
rpc_backend = rabbit	(StrOpt) The messaging driver to use, defaults to rabbit. Other drivers include qpid and zmq.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC greenthread pool.
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1.
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets.
rpc_zmq_matchmaker = oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver.
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port.
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.
[cells]	
rpc_driver_queue_base = cells.intercell	(StrOpt) Base queue name to use when communicating between cells. Various topics by message type will be appended to this.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).
[upgrade_levels]	

Configuration option = Default value	Description
baseapi = None	(StrOpt) Set a version cap for messages sent to the base api in any service

Table 2.47. Description of configuration options for s3

Configuration option = Default value	Description
[DEFAULT]	
buckets_path = \$state_path/buckets	(StrOpt) Path to S3 buckets
image_decryption_dir = /tmp	(StrOpt) Parent directory for tempdir used for image decryption
s3_access_key = notchecked	(StrOpt) Access key to use for S3 server for images
s3_affix_tenant = False	(BoolOpt) Whether to affix the tenant id to the access key when downloading from S3
s3_host = \$my_ip	(StrOpt) Hostname or IP for OpenStack to use when accessing the S3 api
s3_listen = 0.0.0.0	(StrOpt) IP address for S3 API to listen
s3_listen_port = 3333	(IntOpt) Port for S3 API to listen
s3_port = 3333	(IntOpt) Port used when accessing the S3 api
s3_secret_key = notchecked	(StrOpt) Secret key to use for S3 server for images
s3_use_ssl = False	(BoolOpt) Whether to use SSL when talking to S3

Table 2.48. Description of configuration options for scheduling

Configuration option = Default value	Description
[DEFAULT]	
aggregate_image_properties_isolation_namespace = None	(StrOpt) Force the filter to consider only keys matching the given namespace.
aggregate_image_properties_isolation_separator = .	(StrOpt) The separator used between the namespace and keys
cpu_allocation_ratio = 16.0	(FloatOpt) Virtual CPU to physical CPU allocation ratio which affects all CPU filters. This configuration specifies a global ratio for CoreFilter. For AggregateCoreFilter, it will fall back to this configuration value if no per-aggregate setting found.
disk_allocation_ratio = 1.0	(FloatOpt) Virtual disk to physical disk allocation ratio
isolated_hosts =	(ListOpt) Host reserved for specific images
isolated_images =	(ListOpt) Images to run on isolated host
max_instances_per_host = 50	(IntOpt) Ignore hosts that have too many instances
max_io_ops_per_host = 8	(IntOpt) Ignore hosts that have too many builds/resizes/snaps/migrations
ram_allocation_ratio = 1.5	(FloatOpt) Virtual ram to physical ram allocation ratio which affects all ram filters. This configuration specifies a global ratio for RamFilter. For AggregateRamFilter, it will fall back to this configuration value if no per-aggregate setting found.
ram_weight_multiplier = 1.0	(FloatOpt) Multiplier used for weighing ram. Negative numbers mean to stack vs spread.
reserved_host_disk_mb = 0	(IntOpt) Amount of disk in MB to reserve for the host
reserved_host_memory_mb = 512	(IntOpt) Amount of memory in MB to reserve for the host
restrict_isolated_hosts_to_isolated_images = True	(BoolOpt) Whether to force isolated hosts to run only isolated images

Configuration option = Default value	Description
<code>scheduler_available_filters = ['nova.scheduler.filters.all_filters']</code>	(MultiStrOpt) Filter classes available to the scheduler which may be specified more than once. An entry of "nova.scheduler.filters.standard_filters" maps to all filters included with nova.
<code>scheduler_default_filters = RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter</code>	(ListOpt) Which filter class names to use for filtering hosts when not specified in the request.
<code>scheduler_driver = nova.scheduler.filter_scheduler.FilterScheduler</code>	(StrOpt) Default driver to use for the scheduler
<code>scheduler_driver_task_period = 60</code>	(IntOpt) How often (in seconds) to run periodic tasks in the scheduler driver of your choice. Please note this is likely to interact with the value of <code>service_down_time</code> , but exactly how they interact will depend on your choice of scheduler driver.
<code>scheduler_host_manager = nova.scheduler.host_manager.HostManager</code>	(StrOpt) The scheduler host manager class to use
<code>scheduler_host_subset_size = 1</code>	(IntOpt) New instances will be scheduled on a host chosen randomly from a subset of the N best hosts. This property defines the subset size that a host is chosen from. A value of 1 chooses the first host returned by the weighing functions. This value must be at least 1. Any value less than 1 will be ignored, and 1 will be used instead
<code>scheduler_json_config_location =</code>	(StrOpt) Absolute path to scheduler configuration JSON file.
<code>scheduler_manager = nova.scheduler.manager.SchedulerManager</code>	(StrOpt) Full class name for the Manager for scheduler
<code>scheduler_max_attempts = 3</code>	(IntOpt) Maximum number of attempts to schedule an instance
<code>scheduler_topic = scheduler</code>	(StrOpt) The topic scheduler nodes listen on
<code>scheduler_weight_classes = nova.scheduler.weights.all_weighers</code>	(ListOpt) Which weight class names to use for weighing hosts
[cells]	
<code>ram_weight_multiplier = 10.0</code>	(FloatOpt) Multiplier used for weighing ram. Negative numbers mean to stack vs spread.
<code>scheduler_filter_classes = nova.cells.filters.all_filters</code>	(ListOpt) Filter classes the cells scheduler should use. An entry of "nova.cells.filters.all_filters" maps to all cells filters included with nova.
<code>scheduler_retries = 10</code>	(IntOpt) How many retries when no cells are available.
<code>scheduler_retry_delay = 2</code>	(IntOpt) How often to retry in seconds when no cells are available.
<code>scheduler_weight_classes = nova.cells.weights.all_weighers</code>	(ListOpt) Weigher classes the cells scheduler should use. An entry of "nova.cells.weights.all_weighers" maps to all cell weighers included with nova.
[metrics]	
<code>required = True</code>	(BoolOpt) How to treat the unavailable metrics. When a metric is NOT available for a host, if it is set to be True, it would raise an exception, so it is recommended to use the scheduler filter <code>MetricFilter</code> to filter out those hosts. If it is set to be False, the unavailable metric would be treated as a negative factor in weighing process, the returned value would be set by the option <code>weight_of_unavailable</code> .
<code>weight_multiplier = 1.0</code>	(FloatOpt) Multiplier used for weighing metrics.
<code>weight_of_unavailable = -10000.0</code>	(FloatOpt) The final weight value to be returned if <code>required</code> is set to False and any one of the metrics set by <code>weight_setting</code> is unavailable.

Configuration option = Default value	Description
weight_setting =	(ListOpt) How the metrics are going to be weighed. This should be in the form of "<name1>=<ratio1>, <name2>=<ratio2>, ...", where <nameX> is one of the metrics to be weighed, and <ratioX> is the corresponding ratio. So for "name1=1.0, name2=-1.0" The final weight would be name1.value * 1.0 + name2.value * -1.0.

Table 2.49. Description of configuration options for spice

Configuration option = Default value	Description
[spice]	
agent_enabled = True	(BoolOpt) Enable spice guest agent support
enabled = False	(BoolOpt) Enable spice related features
html5proxy_base_url = http://127.0.0.1:6082/spice_auto.html	(StrOpt) Location of spice HTML5 console proxy, in the form "http://127.0.0.1:6082/spice_auto.html"
keymap = en-us	(StrOpt) Keymap for spice
server_listen = 127.0.0.1	(StrOpt) IP address on which instance spice server should listen
server_proxycient_address = 127.0.0.1	(StrOpt) The address to which proxy clients (like nova-spicehtml5proxy) should connect

Table 2.50. Description of configuration options for testing

Configuration option = Default value	Description
[DEFAULT]	
allowed_rpc_exception_modules = oslo.messaging.exceptions, nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.
backdoor_port = None	(StrOpt) 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 it 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 service's log file.
fake_call = False	(BoolOpt) If True, skip using the queue and make local calls
fake_network = False	(BoolOpt) If passed, use fake network devices and addresses
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider.
monkey_patch = False	(BoolOpt) Whether to log monkey patching
monkey_patch_modules = nova.api.ec2.cloud:nova.notifications.notify_decorator, nova.compute.api:nova.notifications.notify_decorator	(ListOpt) List of modules/decorators to monkey patch

Table 2.51. Description of configuration options for tilera

Configuration option = Default value	Description
[baremetal]	
tile_pdu_ip = 10.0.100.1	(StrOpt) IP address of tilera pdu
tile_pdu_mgr = /tftpboot/pdu_mgr	(StrOpt) Management script for tilera pdu
tile_pdu_off = 2	(IntOpt) Power status of tilera PDU is OFF

Configuration option = Default value	Description
tile_pdu_on = 1	(IntOpt) Power status of tilera PDU is ON
tile_pdu_status = 9	(IntOpt) Power status of tilera PDU
tile_power_wait = 9	(IntOpt) Wait time in seconds until check the result after tilera power operations

Table 2.52. Description of configuration options for trustedcomputing

Configuration option = Default value	Description
[trusted_computing]	
attestation_api_url = /OpenAttestationWebServices/V1.0	(StrOpt) Attestation web API URL
attestation_auth_blob = None	(StrOpt) Attestation authorization blob - must change
attestation_auth_timeout = 60	(IntOpt) Attestation status cache valid period length
attestation_port = 8443	(StrOpt) Attestation server port
attestation_server = None	(StrOpt) Attestation server HTTP
attestation_server_ca_file = None	(StrOpt) Attestation server Cert file for Identity verification

Table 2.53. Description of configuration options for upgrade_levels

Configuration option = Default value	Description
[cells]	
scheduler = nova.cells.scheduler.CellsScheduler	(StrOpt) Cells scheduler to use
[upgrade_levels]	
cells = None	(StrOpt) Set a version cap for messages sent to local cells services
cert = None	(StrOpt) Set a version cap for messages sent to cert services
compute = None	(StrOpt) Set a version cap for messages sent to compute services. If you plan to do a live upgrade from havana to icehouse, you should set this option to "icehouse-compat" before beginning the live upgrade procedure.
conductor = None	(StrOpt) Set a version cap for messages sent to conductor services
console = None	(StrOpt) Set a version cap for messages sent to console services
consoleauth = None	(StrOpt) Set a version cap for messages sent to consoleauth services
intercell = None	(StrOpt) Set a version cap for messages sent between cells services
network = None	(StrOpt) Set a version cap for messages sent to network services
scheduler = None	(StrOpt) Set a version cap for messages sent to scheduler services

Table 2.54. Description of configuration options for vmware

Configuration option = Default value	Description
[vmware]	
api_retry_count = 10	(IntOpt) The number of times we retry on failures, e.g., socket error, etc.
cluster_name = None	(MultiStrOpt) Name of a VMware Cluster ComputeResource. Used only if compute_driver is vmwareapi.VMwareVCDriver.

Configuration option = Default value	Description
datastore_regex = None	(StrOpt) Regex to match the name of a datastore.
host_ip = None	(StrOpt) Hostname or IP address for connection to VMware ESX/VC host.
host_password = None	(StrOpt) Password for connection to VMware ESX/VC host.
host_username = None	(StrOpt) Username for connection to VMware ESX/VC host.
integration_bridge = br-int	(StrOpt) Name of Integration Bridge
maximum_objects = 100	(IntOpt) The maximum number of ObjectContent data objects that should be returned in a single result. A positive value will cause the operation to suspend the retrieval when the count of objects reaches the specified maximum. The server may still limit the count to something less than the configured value. Any remaining objects may be retrieved with additional requests.
task_poll_interval = 0.5	(FloatOpt) The interval used for polling of remote tasks.
use_linked_clone = True	(BoolOpt) Whether to use linked clone
wSDL_location = None	(StrOpt) Optional VIM Service WSDL Location e.g http://<server>/vimService.wsdl. Optional over-ride to default location for bug work-arounds

Table 2.55. Description of configuration options for vnc

Configuration option = Default value	Description
[DEFAULT]	
novncproxy_base_url = http://127.0.0.1:6080/vnc_auto.html	(StrOpt) Location of VNC console proxy, in the form "http://127.0.0.1:6080/vnc_auto.html"
vnc_enabled = True	(BoolOpt) Enable VNC related features
vnc_keymap = en-us	(StrOpt) Keymap for VNC
vncserver_listen = 127.0.0.1	(StrOpt) IP address on which instance vncservers should listen
vncserver_proxyclient_address = 127.0.0.1	(StrOpt) The address to which proxy clients (like nova-xvpvncproxy) should connect
[vmware]	
vnc_port = 5900	(IntOpt) VNC starting port
vnc_port_total = 10000	(IntOpt) Total number of VNC ports

Table 2.56. Description of configuration options for volumes

Configuration option = Default value	Description
[DEFAULT]	
cinder_api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to cinder
cinder_ca_certificates_file = None	(StrOpt) Location of ca certificates file to use for cinder client requests.
cinder_catalog_info = volume:cinder:publicURL	(StrOpt) Info to match when looking for cinder in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type>
cinder_cross_az_attach = True	(BoolOpt) Allow attach between instance and volume in different availability zones.
cinder_endpoint_template = None	(StrOpt) Override service catalog lookup with template for cinder endpoint e.g. http://localhost:8776/v1/%(project_id)s

Configuration option = Default value	Description
cinder_http_retries = 3	(IntOpt) Number of cinderclient retries on failed http calls
os_region_name = None	(StrOpt) Region name of this node
volume_api_class = nova.volume.cinder.API	(StrOpt) The full class name of the volume API class to use
volume_usage_poll_interval = 0	(IntOpt) Interval in seconds for gathering volume usages
[baremetal]	
iscsi_iqn_prefix = iqn.2010-10.org.openstack.baremetal	(StrOpt) The iSCSI IQN prefix used in baremetal volume connections.
volume_driver = nova.virt.baremetal.volume_driver.LibvirtVolumeDriver	(StrOpt) Baremetal volume driver.
[hyperv]	
force_volumeutils_v1 = False	(BoolOpt) Force V1 volume utility class
volume_attach_retry_count = 10	(IntOpt) The number of times to retry to attach a volume
volume_attach_retry_interval = 5	(IntOpt) Interval between volume attachment attempts, in seconds
[libvirt]	
glusterfs_mount_point_base = \$state_path/mnt	(StrOpt) Directory where the glusterfs volume is mounted on the compute node
nfs_mount_options = None	(StrOpt) Mount options passed to the NFS client. See section of the nfs man page for details
nfs_mount_point_base = \$state_path/mnt	(StrOpt) Directory where the NFS volume is mounted on the compute node
num_aoe_discover_tries = 3	(IntOpt) Number of times to rediscover AoE target to find volume
num_iscsi_scan_tries = 5	(IntOpt) Number of times to rescan iSCSI target to find volume
num_iscsi_scan_tries = 5	(IntOpt) Number of times to rescan iSER target to find volume
qemu_allowed_storage_drivers =	(ListOpt) Protocols listed here will be accessed directly from QEMU. Currently supported protocols: [gluster]
rbd_secret_uuid = None	(StrOpt) The libvirt UUID of the secret for the rbd_uservolumes
rbd_user = None	(StrOpt) The RADOS client name for accessing rbd volumes
scality_sofs_config = None	(StrOpt) Path or URL to Scality SOFS configuration file
scality_sofs_mount_point = \$state_path/scality	(StrOpt) Base dir where Scality SOFS shall be mounted
[xenserver]	
block_device_creation_timeout = 10	(IntOpt) Time to wait for a block device to be created

Table 2.57. Description of configuration options for vpn

Configuration option = Default value	Description
[DEFAULT]	
boot_script_template = \$pybasedir/nova/cloudpipe/ bootscript.template	(StrOpt) Template for cloudpipe instance boot script
dmz_cidr =	(ListOpt) A list of dmz range that should be accepted
dmz_mask = 255.255.255.0	(StrOpt) Netmask to push into openvpn config
dmz_net = 10.0.0.0	(StrOpt) Network to push into openvpn config
vpn_flavor = m1.tiny	(StrOpt) Flavor for vpn instances
vpn_image_id = 0	(StrOpt) Image ID used when starting up a cloudpipe vpn server
vpn_ip = \$my_ip	(StrOpt) Public IP for the cloudpipe VPN servers

Configuration option = Default value	Description
vpn_key_suffix = -vpn	(StrOpt) Suffix to add to project name for vpn key and secgroups
vpn_start = 1000	(IntOpt) First Vpn port for private networks

Table 2.58. Description of configuration options for wsgi

Configuration option = Default value	Description
[DEFAULT]	
api_paste_config = api-paste.ini	(StrOpt) File name for the paste.deploy config for nova-api
ssl_ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
ssl_cert_file = None	(StrOpt) SSL certificate of API server
ssl_key_file = None	(StrOpt) SSL private key of API server
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
wsgi_default_pool_size = 1000	(IntOpt) Size of the pool of greenthreads used by wsgi
wsgi_log_format = %(client_ip)s "%(request_line)s" status: %(status_code)s len: %(body_length)s time: %(wall_seconds).7f	(StrOpt) A python format string that is used as the template to generate log lines. The following values can be formatted into it: client_ip, date_time, request_line, status_code, body_length, wall_seconds.

Table 2.59. Description of configuration options for xen

Configuration option = Default value	Description
[DEFAULT]	
console_driver = nova.console.xvp.XVPConsoleProxy	(StrOpt) Driver to use for the console proxy
console_vmrc_error_retries = 10	(IntOpt) Number of retries for retrieving VMRC information
console_vmrc_port = 443	(IntOpt) Port for VMware VMRC connections
console_xvp_conf = /etc/xvp.conf	(StrOpt) Generated XVP conf file
console_xvp_conf_template = \$pybasedir/nova/console/xvp.conf.template	(StrOpt) XVP conf template
console_xvp_log = /var/log/xvp.log	(StrOpt) XVP log file
console_xvp_multiplex_port = 5900	(IntOpt) Port for XVP to multiplex VNC connections on
console_xvp_pid = /var/run/xvp.pid	(StrOpt) XVP master process pid file
stub_compute = False	(BoolOpt) Stub calls to compute worker for tests
[libvirt]	
xen_hvmloder_path = /usr/lib/xen/boot/hvmloder	(StrOpt) Location where the Xen hvmloder is kept
[xenserver]	
agent_path = usr/sbin/xen-update-networking	(StrOpt) Specifies the path in which the XenAPI guest agent should be located. If the agent is present, network configuration is not injected into the image. Used if compute_driver=xenapi.XenAPIDriver and flat_injected=True
agent_resetnetwork_timeout = 60	(IntOpt) Number of seconds to wait for agent reply to resetnetwork request
agent_timeout = 30	(IntOpt) Number of seconds to wait for agent reply
agent_version_timeout = 300	(IntOpt) Number of seconds to wait for agent to be fully operational
cache_images = all	(StrOpt) Cache glance images locally. `all` will cache all images, `some` will only cache images that have the

Configuration option = Default value	Description
	image_property `cache_in_nova=True`, and `none` turns off caching entirely
check_host = True	(BoolOpt) Ensure compute service is running on host XenAPI connects to.
connection_concurrent = 5	(IntOpt) Maximum number of concurrent XenAPI connections. Used only if compute_driver=xenapi.XenAPIDriver
connection_password = None	(StrOpt) Password for connection to XenServer/Xen Cloud Platform. Used only if compute_driver=xenapi.XenAPIDriver
connection_url = None	(StrOpt) URL for connection to XenServer/Xen Cloud Platform. A special value of unix://local can be used to connect to the local unix socket. Required if compute_driver=xenapi.XenAPIDriver
connection_username = root	(StrOpt) Username for connection to XenServer/Xen Cloud Platform. Used only if compute_driver=xenapi.XenAPIDriver
default_os_type = linux	(StrOpt) Default OS type
disable_agent = False	(BoolOpt) Disables the use of the XenAPI agent in any image regardless of what image properties are present.
image_compression_level = None	(IntOpt) Compression level for images, e.g., 9 for gzip -9. Range is 1-9, 9 being most compressed but most CPU intensive on dom0.
image_upload_handler = nova.virt.xenapi.image.glance.GlanceStore	(StrOpt) Dom0 plugin driver used to handle image uploads.
introduce_vdi_retry_wait = 20	(IntOpt) Number of seconds to wait for an SR to settle if the VDI does not exist when first introduced
ipxe_boot_menu_url = None	(StrOpt) URL to the iPXE boot menu
ipxe_mkisofs_cmd = mkisofs	(StrOpt) Name and optionally path of the tool used for ISO image creation
ipxe_network_name = None	(StrOpt) Name of network to use for booting iPXE ISOs
iqn_prefix = iqn.2010-10.org.openstack	(StrOpt) IQN Prefix
login_timeout = 10	(IntOpt) Timeout in seconds for XenAPI login.
max_kernel_ramdisk_size = 16777216	(IntOpt) Maximum size in bytes of kernel or ramdisk images
num_vbd_unplug_retries = 10	(IntOpt) Maximum number of retries to unplug VBD
ovs_integration_bridge = xapi1	(StrOpt) Name of Integration Bridge used by Open vSwitch
remap_vbd_dev = False	(BoolOpt) Used to enable the remapping of VBD dev (Works around an issue in Ubuntu Maverick)
remap_vbd_dev_prefix = sd	(StrOpt) Specify prefix to remap VBD dev to (ex. /dev/xvdb -> /dev/sdb)
running_timeout = 60	(IntOpt) Number of seconds to wait for instance to go to running state
sparse_copy = True	(BoolOpt) Whether to use sparse_copy for copying data on a resize down (False will use standard dd). This speeds up resizes down considerably since large runs of zeros won't have to be rsynced
sr_base_path = /var/run/sr-mount	(StrOpt) Base path to the storage repository
sr_matching_filter = default-sr:true	(StrOpt) Filter for finding the SR to be used to install guest instances on. To use the Local Storage in default XenServer/XCP installations set this flag to other-config:i18n-key=local-storage. To select an SR with a different matching criteria, you could set it to other-

Configuration option = Default value	Description
	config:my_favorite_sr=true. On the other hand, to fall back on the Default SR, as displayed by XenCenter, set this flag to: default-sr=true
target_host = None	(StrOpt) The iSCSI Target Host
target_port = 3260	(StrOpt) The iSCSI Target Port, default is port 3260
torrent_base_url = None	(StrOpt) Base URL for torrent files.
torrent_download_stall_cutoff = 600	(IntOpt) Number of seconds a download can remain at the same progress percentage w/o being considered a stall
torrent_images = none	(StrOpt) Whether or not to download images via Bit Torrent (all some none).
torrent_listen_port_end = 6891	(IntOpt) End of port range to listen on
torrent_listen_port_start = 6881	(IntOpt) Beginning of port range to listen on
torrent_max_last_accessed = 86400	(IntOpt) Cached torrent files not accessed within this number of seconds can be reaped
torrent_max_seeder_processes_per_host = 1	(IntOpt) Maximum number of seeder processes to run concurrently within a given dom0. (-1 = no limit)
torrent_seed_chance = 1.0	(FloatOpt) Probability that peer will become a seeder. (1.0 = 100%)
torrent_seed_duration = 3600	(IntOpt) Number of seconds after downloading an image via BitTorrent that it should be seeded for other peers.
use_agent_default = False	(BoolOpt) Determines if the XenAPI agent should be used when the image used does not contain a hint to declare if the agent is present or not. The hint is a glance property "xenapi_use_agent" that has the value "True" or "False". Note that waiting for the agent when it is not present will significantly increase server boot times.
use_join_force = True	(BoolOpt) To use for hosts with different CPUs
vhd_coalesce_max_attempts = 20	(IntOpt) Max number of times to poll for VHD to coalesce. Used only if compute_driver=xenapi.XenAPIDriver
vhd_coalesce_poll_interval = 5.0	(FloatOpt) The interval used for polling of coalescing vhds. Used only if compute_driver=xenapi.XenAPIDriver

Table 2.60. Description of configuration options for xvpvncproxy

Configuration option = Default value	Description
[DEFAULT]	
xvpvncproxy_base_url = http://127.0.0.1:6081/console	(StrOpt) Location of nova xvp VNC console proxy, in the form "http://127.0.0.1:6081/console"
xvpvncproxy_host = 0.0.0.0	(StrOpt) Address that the XCP VNC proxy should bind to
xvpvncproxy_port = 6081	(IntOpt) Port that the XCP VNC proxy should bind to

Table 2.61. Description of configuration options for zookeeper

Configuration option = Default value	Description
[zookeeper]	
address = None	(StrOpt) The ZooKeeper addresses for servicegroup service in the format of host1:port,host2:port,host3:port
recv_timeout = 4000	(IntOpt) The recv_timeout parameter for the zk session
sg_prefix = /servicegroups	(StrOpt) The prefix used in ZooKeeper to store ephemeral nodes
sg_retry_interval = 5	(IntOpt) Number of seconds to wait until retrying to join the session

Additional sample configuration files

Files in this section can be found in `/etc/nova`.

api-paste.ini

The Compute service stores its API configuration settings in the `api-paste.ini` file.

```
#####  
# Metadata #  
#####  
[composite:metadata]  
use = egg:Paste#urlmap  
/: meta  
  
[pipeline:meta]  
pipeline = ec2faultwrap logrequest metaapp  
  
[app:metaapp]  
paste.app_factory = nova.api.metadata.handler:MetadataRequestHandler.factory  
  
#####  
# EC2 #  
#####  
  
[composite:ec2]  
use = egg:Paste#urlmap  
/services/Cloud: ec2cloud  
  
[composite:ec2cloud]  
use = call:nova.api.auth:pipeline_factory  
noauth = ec2faultwrap logrequest ec2noauth cloudrequest validator ec2executor  
keystone = ec2faultwrap logrequest ec2keystoneauth cloudrequest validator  
ec2executor  
  
[filter:ec2faultwrap]  
paste.filter_factory = nova.api.ec2:FaultWrapper.factory  
  
[filter:logrequest]  
paste.filter_factory = nova.api.ec2:RequestLogging.factory  
  
[filter:ec2lockout]  
paste.filter_factory = nova.api.ec2:Lockout.factory  
  
[filter:ec2keystoneauth]  
paste.filter_factory = nova.api.ec2:EC2KeystoneAuth.factory  
  
[filter:ec2noauth]  
paste.filter_factory = nova.api.ec2:NoAuth.factory  
  
[filter:cloudrequest]  
controller = nova.api.ec2.cloud.CloudController  
paste.filter_factory = nova.api.ec2:Requestify.factory  
  
[filter:authorizer]  
paste.filter_factory = nova.api.ec2:Authorizer.factory  
  
[filter:validator]
```



```
[filter:keystonecontext]
paste.filter_factory = nova.api.auth:NovaKeystoneContext.factory

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
```

policy.json

The `policy.json` file defines additional access controls that apply to the Compute service.

```
{
  "context_is_admin": "role:admin",
  "admin_or_owner": "is_admin:True or project_id:%(project_id)s",
  "default": "rule:admin_or_owner",

  "cells_scheduler_filter:TargetCellFilter": "is_admin:True",

  "compute:create": "",
  "compute:create:attach_network": "",
  "compute:create:attach_volume": "",
  "compute:create:forced_host": "is_admin:True",
  "compute:get_all": "",
  "compute:get_all_tenants": "",
  "compute:start": "rule:admin_or_owner",
  "compute:stop": "rule:admin_or_owner",
  "compute:unlock_override": "rule:admin_api",

  "compute:shelve": "",
  "compute:shelve_offload": "",
  "compute:unshelve": "",

  "compute:volume_snapshot_create": "",
  "compute:volume_snapshot_delete": "",

  "admin_api": "is_admin:True",
  "compute:v3:servers:start": "rule:admin_or_owner",
  "compute:v3:servers:stop": "rule:admin_or_owner",
  "compute_extension:v3:os-access-ips:discoverable": "",
  "compute_extension:v3:os-access-ips": "",
  "compute_extension:accounts": "rule:admin_api",
  "compute_extension:admin_actions": "rule:admin_api",
  "compute_extension:admin_actions:pause": "rule:admin_or_owner",
  "compute_extension:admin_actions:unpause": "rule:admin_or_owner",
  "compute_extension:admin_actions:suspend": "rule:admin_or_owner",
  "compute_extension:admin_actions:resume": "rule:admin_or_owner",
  "compute_extension:admin_actions:lock": "rule:admin_or_owner",
  "compute_extension:admin_actions:unlock": "rule:admin_or_owner",
  "compute_extension:admin_actions:resetNetwork": "rule:admin_api",
  "compute_extension:admin_actions:injectNetworkInfo": "rule:admin_api",
  "compute_extension:admin_actions:createBackup": "rule:admin_or_owner",
  "compute_extension:admin_actions:migrateLive": "rule:admin_api",
  "compute_extension:admin_actions:resetState": "rule:admin_api",
  "compute_extension:admin_actions:migrate": "rule:admin_api",
  "compute_extension:v3:os-admin-actions": "rule:admin_api",
  "compute_extension:v3:os-admin-actions:discoverable": "",
```



```
"compute_extension:keypairs": "",
"compute_extension:keypairs:index": "",
"compute_extension:keypairs:show": "",
"compute_extension:keypairs:create": "",
"compute_extension:keypairs:delete": "",
"compute_extension:v3:keypairs:discoverable": "",
"compute_extension:v3:keypairs": "",
"compute_extension:v3:keypairs:index": "",
"compute_extension:v3:keypairs:show": "",
"compute_extension:v3:keypairs:create": "",
"compute_extension:v3:keypairs:delete": "",
"compute_extension:v3:os-lock-server:discoverable": "",
"compute_extension:v3:os-lock-server:lock": "rule:admin_or_owner",
"compute_extension:v3:os-lock-server:unlock": "rule:admin_or_owner",
"compute_extension:v3:os-migrate-server:discoverable": "",
"compute_extension:v3:os-migrate-server:migrate": "rule:admin_api",
"compute_extension:v3:os-migrate-server:migrate_live": "rule:admin_api",
"compute_extension:multinic": "",
"compute_extension:v3:os-multinic": "",
"compute_extension:v3:os-multinic:discoverable": "",
"compute_extension:networks": "rule:admin_api",
"compute_extension:networks:view": "",
"compute_extension:networks_associate": "rule:admin_api",
"compute_extension:v3:os-pause-server:discoverable": "",
"compute_extension:v3:os-pause-server:pause": "rule:admin_or_owner",
"compute_extension:v3:os-pause-server:unpause": "rule:admin_or_owner",
"compute_extension:v3:os-pci:pci_servers": "",
"compute_extension:v3:os-pci:discoverable": "",
"compute_extension:v3:os-pci:index": "rule:admin_api",
"compute_extension:v3:os-pci:detail": "rule:admin_api",
"compute_extension:v3:os-pci:show": "rule:admin_api",
"compute_extension:quotas:show": "",
"compute_extension:quotas:update": "rule:admin_api",
"compute_extension:quotas:delete": "rule:admin_api",
"compute_extension:v3:os-quota-sets:discoverable": "",
"compute_extension:v3:os-quota-sets:show": "",
"compute_extension:v3:os-quota-sets:update": "rule:admin_api",
"compute_extension:v3:os-quota-sets:delete": "rule:admin_api",
"compute_extension:v3:os-quota-sets:detail": "rule:admin_api",
"compute_extension:quota_classes": "",
"compute_extension:rescue": "",
"compute_extension:v3:os-rescue": "",
"compute_extension:v3:os-rescue:discoverable": "",
"compute_extension:v3:os-scheduler-hints:discoverable": "",
"compute_extension:security_group_default_rules": "rule:admin_api",
"compute_extension:security_groups": "",
"compute_extension:v3:os-security-groups": "",
"compute_extension:v3:os-security-groups:discoverable": "",
"compute_extension:server_diagnostics": "rule:admin_api",
"compute_extension:v3:os-server-diagnostics": "rule:admin_api",
"compute_extension:v3:os-server-diagnostics:discoverable": "",
"compute_extension:server_groups": "",
"compute_extension:server_password": "",
"compute_extension:v3:os-server-password": "",
"compute_extension:v3:os-server-password:discoverable": "",
"compute_extension:server_usage": "",
"compute_extension:v3:os-server-usage": "",
"compute_extension:v3:os-server-usage:discoverable": "",
"compute_extension:services": "rule:admin_api",
"compute_extension:v3:os-services": "rule:admin_api",
```

```
"compute_extension:v3:os-services:discoverable": "",
"compute_extension:v3:server-metadata:discoverable": "",
"compute_extension:v3:servers:discoverable": "",
"compute_extension:shelve": "",
"compute_extension:shelveOffload": "rule:admin_api",
"compute_extension:v3:os-shelve:shelve": "",
"compute_extension:v3:os-shelve:shelve:discoverable": "",
"compute_extension:v3:os-shelve:shelve_offload": "rule:admin_api",
"compute_extension:simple_tenant_usage:show": "rule:admin_or_owner",
"compute_extension:v3:os-suspend-server:discoverable": "",
"compute_extension:v3:os-suspend-server:suspend": "rule:admin_or_owner",
"compute_extension:v3:os-suspend-server:resume": "rule:admin_or_owner",
"compute_extension:simple_tenant_usage:list": "rule:admin_api",
"compute_extension:unshelve": "",
"compute_extension:v3:os-shelve:unshelve": "",
"compute_extension:users": "rule:admin_api",
"compute_extension:v3:os-user-data:discoverable": "",
"compute_extension:virtual_interfaces": "",
"compute_extension:virtual_storage_arrays": "",
"compute_extension:volumes": "",
"compute_extension:volume_attachments:index": "",
"compute_extension:volume_attachments:show": "",
"compute_extension:volume_attachments:create": "",
"compute_extension:volume_attachments:update": "",
"compute_extension:volume_attachments:delete": "",
"compute_extension:volumetypes": "",
"compute_extension:availability_zone:list": "",
"compute_extension:v3:os-availability-zone:list": "",
"compute_extension:v3:os-availability-zone:discoverable": "",
"compute_extension:availability_zone:detail": "rule:admin_api",
"compute_extension:v3:os-availability-zone:detail": "rule:admin_api",
"compute_extension:used_limits_for_admin": "rule:admin_api",
"compute_extension:migrations:index": "rule:admin_api",
"compute_extension:v3:os-migrations:index": "rule:admin_api",
"compute_extension:v3:os-migrations:discoverable": "",
"compute_extension:os-assisted-volume-snapshots:create": "rule:admin_api",
"compute_extension:os-assisted-volume-snapshots:delete": "rule:admin_api",
"compute_extension:console_auth_tokens": "rule:admin_api",
"compute_extension:v3:os-console-auth-tokens": "rule:admin_api",
"compute_extension:os-server-external-events:create": "rule:admin_api",
"compute_extension:v3:os-server-external-events:create": "rule:admin_api",

"volume:create": "",
"volume:get_all": "",
"volume:get_volume_metadata": "",
"volume:get_snapshot": "",
"volume:get_all_snapshots": "",

"volume_extension:types_manage": "rule:admin_api",
"volume_extension:types_extra_specs": "rule:admin_api",
"volume_extension:volume_admin_actions:reset_status": "rule:admin_api",
"volume_extension:snapshot_admin_actions:reset_status": "rule:admin_api",
"volume_extension:volume_admin_actions:force_delete": "rule:admin_api",

"network:get_all": "",
"network:get": "",
"network:create": "",
"network:delete": "",
```

```
"network:associate": "",
"network:disassociate": "",
"network:get_vifs_by_instance": "",
"network:allocate_for_instance": "",
"network:deallocate_for_instance": "",
"network:validate_networks": "",
"network:get_instance_uuids_by_ip_filter": "",
"network:get_instance_id_by_floating_address": "",
"network:setup_networks_on_host": "",
"network:get_backdoor_port": "",

"network:get_floating_ip": "",
"network:get_floating_ip_pools": "",
"network:get_floating_ip_by_address": "",
"network:get_floating_ips_by_project": "",
"network:get_floating_ips_by_fixed_address": "",
"network:allocate_floating_ip": "",
"network:deallocate_floating_ip": "",
"network:associate_floating_ip": "",
"network:disassociate_floating_ip": "",
"network:release_floating_ip": "",
"network:migrate_instance_start": "",
"network:migrate_instance_finish": "",

"network:get_fixed_ip": "",
"network:get_fixed_ip_by_address": "",
"network:add_fixed_ip_to_instance": "",
"network:remove_fixed_ip_from_instance": "",
"network:add_network_to_project": "",
"network:get_instance_nw_info": "",

"network:get_dns_domains": "",
"network:add_dns_entry": "",
"network:modify_dns_entry": "",
"network:delete_dns_entry": "",
"network:get_dns_entries_by_address": "",
"network:get_dns_entries_by_name": "",
"network:create_private_dns_domain": "",
"network:create_public_dns_domain": "",
"network:delete_dns_domain": ""
}
```

rootwrap.conf

The `rootwrap.conf` file defines configuration values used by the `rootwrap` script when the Compute service needs to escalate its privileges to those of the root user.

```
        # Configuration for nova-rootwrap
# This file should be owned by (and only-writeable by) the root user

[DEFAULT]
# List of directories to load filter definitions from (separated by ',').
# These directories MUST all be only writeable by root !
filters_path=/etc/nova/rootwrap.d,/usr/share/nova/rootwrap

# List of directories to search executables in, in case filters do not
# explicitly specify a full path (separated by ',')
```

```
# If not specified, defaults to system PATH environment variable.
# These directories MUST all be only writeable by root !
exec_dirs=/sbin,/usr/sbin,/bin,/usr/bin

# Enable logging to syslog
# Default value is False
use_syslog=False

# Which syslog facility to use.
# Valid values include auth, authpriv, syslog, user0, user1...
# Default value is 'syslog'
syslog_log_facility=syslog

# Which messages to log.
# INFO means log all usage
# ERROR means only log unsuccessful attempts
syslog_log_level=ERROR
```

New, updated and deprecated options for nova

Table 2.62. New options

Option = default value	(Type) Help string
aggregate_image_properties_isolation_namespace = None	(StrOpt) Force the filter to consider only keys matching the given namespace.
aggregate_image_properties_isolation_separator = .	(StrOpt) The separator used between the namespace and keys
compute_available_monitors = ['nova.compute.monitors.all_monitors']	(MultiStrOpt) Monitor classes available to the compute which may be specified more than once.
compute_monitors = []	(ListOpt) A list of monitors that can be used for getting compute metrics.
image_cache_subdirectory_name = _base	(StrOpt) Where cached images are stored under \$instances_path. This is NOT the full path - just a folder name. For per-compute-host cached images, set to _base_\$my_ip
image_handlers = ['download']	(ListOpt) Specifies which image handler extension names to use for handling images. The first extension in the list which can handle the image with a suitable location will be used.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
neutron_admin_tenant_id = None	(StrOpt) Tenant id for connecting to neutron in admin context
ovs_vsctl_timeout = 120	(IntOpt) Amount of time, in seconds, that ovs_vsctl should wait for a response from the database. 0 is to wait forever.

Option = default value	(Type) Help string
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
scheduler_driver_task_period = 60	(IntOpt) How often (in seconds) to run periodic tasks in the scheduler driver of your choice. Please note this is likely to interact with the value of service_down_time, but exactly how they interact will depend on your choice of scheduler driver.
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
vif_plugging_is_fatal = True	(BoolOpt) Fail instance boot if vif plugging fails
vif_plugging_timeout = 300	(IntOpt) Number of seconds to wait for neutron vif plugging events to arrive before continuing or failing (see vif_plugging_is_fatal). If this is set to zero and vif_plugging_is_fatal is False, events should not be expected to arrive at all.
wsgi_default_pool_size = 1000	(IntOpt) Size of the pool of greenthreads used by wsgi
baremetal/flavor_extra_specs = []	(ListOpt) A list of additional capabilities corresponding to flavor_extra_specs for this compute host to advertise. Valid entries are name=value, pairs For example, "key1:val1, key2:val2"
baremetal/use_file_injection = False	(BoolOpt) If True, enable file injection for network info, files and admin password
cells/offset_weight_multiplier = 1.0	(FloatOpt) Multiplier used to weigh offset weigher.
database/db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
database/db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
database/db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
database/db_retry_interval = 1	(IntOpt) seconds between db connection retries
database/mysql_sql_mode = None	(StrOpt) The SQL mode to be used for MySQL sessions (default is empty, meaning do not override any server-side SQL mode setting)
database/sqlite_db = nova.sqlite	(StrOpt) The file name to use with SQLite
database/sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
database/use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost
libvirt/block_migration_flag = VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_NON_SHARED_INC	(StrOpt) Migration flags to be set for block migration
libvirt/checksum_base_images = False	(BoolOpt) Write a checksum for files in _base to disk
libvirt/checksum_interval_seconds = 3600	(IntOpt) How frequently to checksum base images
libvirt/connection_uri =	(StrOpt) Override the default libvirt URI (which is dependent on virt_type)
libvirt/cpu_mode = None	(StrOpt) Set to "host-model" to clone the host CPU feature flags; to "host-passthrough" to use the host CPU model

Option = default value	(Type) Help string
libvirt/cpu_model = None	(StrOpt) Set to a named libvirt CPU model (see names listed in /usr/share/libvirt/cpu_map.xml). Only has effect if cpu_mode="custom" and virt_type="kvm qemu"
libvirt/disk_cachemodes = []	(ListOpt) Specific cachemodes to use for different disk types e.g: file=directsync,block=none
libvirt/disk_prefix = None	(StrOpt) Override the default disk prefix for the devices attached to a server, which is dependent on virt_type. (valid options are: sd, xvd, uvd, vd)
libvirt/glusterfs_mount_point_base = \$state_path/mnt	(StrOpt) Directory where the glusterfs volume is mounted on the compute node
libvirt/image_info_filename_pattern = \$instances_path/\$image_cache_subdirectory_name/(image)s.info	(StrOpt) Allows image information files to be stored in non-standard locations
libvirt/images_rbd_ceph_conf =	(StrOpt) Path to the ceph configuration file to use
libvirt/images_rbd_pool = rbd	(StrOpt) The RADOS pool in which rbd volumes are stored
libvirt/images_type = default	(StrOpt) VM Images format. Acceptable values are: raw, qcow2, lvm, rbd, default. If default is specified, then use_cow_images flag is used instead of this one.
libvirt/images_volume_group = None	(StrOpt) LVM Volume Group that is used for VM images, when you specify images_type=lvm.
libvirt/inject_key = False	(BoolOpt) Inject the ssh public key at boot time
libvirt/inject_partition = -2	(IntOpt) The partition to inject to : -2 => disable, -1 => inspect (libguestfs only), 0 => not partitioned, >0 => partition number
libvirt/inject_password = False	(BoolOpt) Inject the admin password at boot time, without an agent.
libvirt/iscsi_use_multipath = False	(BoolOpt) Use multipath connection of the iSCSI volume
libvirt/iser_use_multipath = False	(BoolOpt) Use multipath connection of the iSER volume
libvirt/live_migration_bandwidth = 0	(IntOpt) Maximum bandwidth to be used during migration, in Mbps
libvirt/live_migration_flag = VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER	(StrOpt) Migration flags to be set for live migration
libvirt/live_migration_uri = qemu+tcp://%s/system	(StrOpt) Migration target URI (any included "%s" is replaced with the migration target hostname)
libvirt/nfs_mount_options = None	(StrOpt) Mount options passed to the NFS client. See section of the nfs man page for details
libvirt/nfs_mount_point_base = \$state_path/mnt	(StrOpt) Directory where the NFS volume is mounted on the compute node
libvirt/num_aoe_discover_tries = 3	(IntOpt) Number of times to rediscover AoE target to find volume
libvirt/num_iscsi_scan_tries = 5	(IntOpt) Number of times to rescan iSCSI target to find volume
libvirt/num_iser_scan_tries = 5	(IntOpt) Number of times to rescan iSER target to find volume
libvirt/qemu_allowed_storage_drivers = []	(ListOpt) Protocols listed here will be accessed directly from QEMU. Currently supported protocols: [gluster]
libvirt/rbd_secret_uuid = None	(StrOpt) The libvirt UUID of the secret for the rbd_uservolumes
libvirt/rbd_user = None	(StrOpt) The RADOS client name for accessing rbd volumes

Option = default value	(Type) Help string
libvirt/remove_unused_kernels = False	(BoolOpt) Should unused kernel images be removed? This is only safe to enable if all compute nodes have been updated to support this option. This will be enabled by default in future.
libvirt/remove_unused_resized_minimum_age_seconds = 3600	(IntOpt) Unused resized base images younger than this will not be removed
libvirt/rescue_image_id = None	(StrOpt) Rescue ami image
libvirt/rescue_kernel_id = None	(StrOpt) Rescue aki image
libvirt/rescue_ramdisk_id = None	(StrOpt) Rescue ari image
libvirt/rng_dev_path = None	(StrOpt) A path to a device that will be used as source of entropy on the host. Permitted options are: /dev/random or /dev/hwrng
libvirt/scality_sofs_config = None	(StrOpt) Path or URL to Scality SOFS configuration file
libvirt/scality_sofs_mount_point = \$state_path/scality	(StrOpt) Base dir where Scality SOFS shall be mounted
libvirt/snapshot_compression = False	(BoolOpt) Compress snapshot images when possible. This currently applies exclusively to qcow2 images
libvirt/snapshot_image_format = None	(StrOpt) Snapshot image format (valid options are : raw, qcow2, vmdk, vdi). Defaults to same as source image
libvirt/snapshots_directory = \$instances_path/snapshots	(StrOpt) Location where libvirt driver will store snapshots before uploading them to image service
libvirt/sparse_logical_volumes = False	(BoolOpt) Create sparse logical volumes (with virtulsize) if this flag is set to True.
libvirt/use_usb_tablet = True	(BoolOpt) Sync virtual and real mouse cursors in Windows VMs
libvirt/use_virtio_for_bridges = True	(BoolOpt) Use virtio for bridge interfaces with KVM/QEMU
libvirt/vif_driver = nova.virt.libvirt.vif.LibvirtGenericVIFDriver	(StrOpt) DEPRECATED. The libvirt VIF driver to configure the VIFs. This option is deprecated and will be removed in the Juno release.
libvirt/virt_type = kvm	(StrOpt) Libvirt domain type (valid options are: kvm, lxc, qemu, uml, xen)
libvirt/volume_clear = zero	(StrOpt) Method used to wipe old volumes (valid options are: none, zero, shred)
libvirt/volume_clear_size = 0	(IntOpt) Size in MiB to wipe at start of old volumes. 0 => all
libvirt/volume_drivers = ['iscsi=nova.virt.libvirt.volume.LibvirtISCSIVolumeDriver', 'iser=nova.virt.libvirt.volume.LibvirtISERVVolumeDriver', 'local=nova.virt.libvirt.volume.LibvirtVolumeDriver', 'fake=nova.virt.libvirt.volume.LibvirtFakeVolumeDriver', 'rbd=nova.virt.libvirt.volume.LibvirtNetVolumeDriver', 'sheepdog=nova.virt.libvirt.volume.LibvirtNetVolumeDriver', 'nfs=nova.virt.libvirt.volume.LibvirtNFSSVolumeDriver', 'aoe=nova.virt.libvirt.volume.LibvirtAOEVVolumeDriver', 'glusterfs=nova.virt.libvirt.volume.LibvirtGlusterfsVolumeDriver', 'fibre_channel=nova.virt.libvirt.volume.LibvirtFibreChannelVolumeDriver', 'scality=nova.virt.libvirt.volume.LibvirtScalityVolumeDriver']	(ListOpt) Libvirt handlers for remote volumes.
libvirt/wait_soft_reboot_seconds = 120	(IntOpt) Number of seconds to wait for instance to shut down after soft reboot request is made. We fall back to hard reboot if instance does not shutdown within this window.
libvirt/xen_hvmloder_path = /usr/lib/xen/boot/hvmloder	(StrOpt) Location where the Xen hvmloder is kept
metrics/required = True	(BoolOpt) How to treat the unavailable metrics. When a metric is NOT available for a host, if it is set to be True, it

Option = default value	(Type) Help string
	would raise an exception, so it is recommended to use the scheduler filter MetricFilter to filter out those hosts. If it is set to be False, the unavailable metric would be treated as a negative factor in weighing process, the returned value would be set by the option weight_of_unavailable.
metrics/weight_multiplier = 1.0	(FloatOpt) Multiplier used for weighing metrics.
metrics/weight_of_unavailable = -10000.0	(FloatOpt) The final weight value to be returned if required is set to False and any one of the metrics set by weight_setting is unavailable.
metrics/weight_setting = []	(ListOpt) How the metrics are going to be weighed. This should be in the form of "<name1>=<ratio1>, <name2>=<ratio2>, ...", where <nameX> is one of the metrics to be weighed, and <ratioX> is the corresponding ratio. So for "name1=1.0, name2=-1.0" The final weight would be name1.value * 1.0 + name2.value * -1.0.
rdp/enabled = False	(BoolOpt) Enable RDP related features
rdp/html5_proxy_base_url = http://127.0.0.1:6083/	(StrOpt) Location of RDP html5 console proxy, in the form "http://127.0.0.1:6083/"
xenserver/agent_path = usr/sbin/xen-update-networking	(StrOpt) Specifies the path in which the XenAPI guest agent should be located. If the agent is present, network configuration is not injected into the image. Used if compute_driver=xenapi.XenAPIDriver and flat_injected=True
xenserver/agent_resetnetwork_timeout = 60	(IntOpt) Number of seconds to wait for agent reply to resetnetwork request
xenserver/agent_timeout = 30	(IntOpt) Number of seconds to wait for agent reply
xenserver/agent_version_timeout = 300	(IntOpt) Number of seconds to wait for agent to be fully operational
xenserver/block_device_creation_timeout = 10	(IntOpt) Time to wait for a block device to be created
xenserver/cache_images = all	(StrOpt) Cache glance images locally. `all` will cache all images, `some` will only cache images that have the image_property `cache_in_nova=True`, and `none` turns off caching entirely
xenserver/check_host = True	(BoolOpt) Ensure compute service is running on host XenAPI connects to.
xenserver/connection_concurrent = 5	(IntOpt) Maximum number of concurrent XenAPI connections. Used only if compute_driver=xenapi.XenAPIDriver
xenserver/connection_password = None	(StrOpt) Password for connection to XenServer/Xen Cloud Platform. Used only if compute_driver=xenapi.XenAPIDriver
xenserver/connection_url = None	(StrOpt) URL for connection to XenServer/Xen Cloud Platform. A special value of unix://local can be used to connect to the local unix socket. Required if compute_driver=xenapi.XenAPIDriver
xenserver/connection_username = root	(StrOpt) Username for connection to XenServer/Xen Cloud Platform. Used only if compute_driver=xenapi.XenAPIDriver
xenserver/default_os_type = linux	(StrOpt) Default OS type
xenserver/disable_agent = False	(BoolOpt) Disables the use of the XenAPI agent in any image regardless of what image properties are present.
xenserver/image_compression_level = None	(IntOpt) Compression level for images, e.g., 9 for gzip -9. Range is 1-9, 9 being most compressed but most CPU intensive on dom0.
xenserver/image_upload_handler = nova.virt.xenapi.image.glance.GlanceStore	(StrOpt) Dom0 plugin driver used to handle image uploads.

Option = default value	(Type) Help string
xenserver/introduce_vdi_retry_wait = 20	(IntOpt) Number of seconds to wait for an SR to settle if the VDI does not exist when first introduced
xenserver/ipxe_boot_menu_url = None	(StrOpt) URL to the iPXE boot menu
xenserver/ipxe_mkisofs_cmd = mkisofs	(StrOpt) Name and optionally path of the tool used for ISO image creation
xenserver/ipxe_network_name = None	(StrOpt) Name of network to use for booting iPXE ISOs
xenserver/iqn_prefix = iqn.2010-10.org.openstack	(StrOpt) IQN Prefix
xenserver/login_timeout = 10	(IntOpt) Timeout in seconds for XenAPI login.
xenserver/max_kernel_ramdisk_size = 16777216	(IntOpt) Maximum size in bytes of kernel or ramdisk images
xenserver/num_vbd_unplug_retries = 10	(IntOpt) Maximum number of retries to unplug VBD
xenserver/ovs_integration_bridge = xapi1	(StrOpt) Name of Integration Bridge used by Open vSwitch
xenserver/remap_vbd_dev = False	(BoolOpt) Used to enable the remapping of VBD dev (Works around an issue in Ubuntu Maverick)
xenserver/remap_vbd_dev_prefix = sd	(StrOpt) Specify prefix to remap VBD dev to (ex. /dev/xvdb -> /dev/sdb)
xenserver/running_timeout = 60	(IntOpt) Number of seconds to wait for instance to go to running state
xenserver/sparse_copy = True	(BoolOpt) Whether to use sparse_copy for copying data on a resize down (False will use standard dd). This speeds up resizes down considerably since large runs of zeros won't have to be rsynced
xenserver/sr_base_path = /var/run/sr-mount	(StrOpt) Base path to the storage repository
xenserver/sr_matching_filter = default-sr:true	(StrOpt) Filter for finding the SR to be used to install guest instances on. To use the Local Storage in default XenServer/XCP installations set this flag to other-config:i18n-key=local-storage. To select an SR with a different matching criteria, you could set it to other-config:my_favorite_sr=true. On the other hand, to fall back on the Default SR, as displayed by XenCenter, set this flag to: default-sr:true
xenserver/target_host = None	(StrOpt) The iSCSI Target Host
xenserver/target_port = 3260	(StrOpt) The iSCSI Target Port, default is port 3260
xenserver/torrent_base_url = None	(StrOpt) Base URL for torrent files.
xenserver/torrent_download_stall_cutoff = 600	(IntOpt) Number of seconds a download can remain at the same progress percentage w/o being considered a stall
xenserver/torrent_images = none	(StrOpt) Whether or not to download images via Bit Torrent (all some none).
xenserver/torrent_listen_port_end = 6891	(IntOpt) End of port range to listen on
xenserver/torrent_listen_port_start = 6881	(IntOpt) Beginning of port range to listen on
xenserver/torrent_max_last_accessed = 86400	(IntOpt) Cached torrent files not accessed within this number of seconds can be reaped
xenserver/torrent_max_seeder_processes_per_host = 1	(IntOpt) Maximum number of seeder processes to run concurrently within a given dom0. (-1 = no limit)
xenserver/torrent_seed_chance = 1.0	(FloatOpt) Probability that peer will become a seeder. (1.0 = 100%)
xenserver/torrent_seed_duration = 3600	(IntOpt) Number of seconds after downloading an image via BitTorrent that it should be seeded for other peers.
xenserver/use_agent_default = False	(BoolOpt) Determines if the XenAPI agent should be used when the image used does not contain a hint to declare if the agent is present or not. The hint is a glance property

Option = default value	(Type) Help string
xenserver/use_join_force = True	(BoolOpt) To use for hosts with different CPUs
xenserver/vhd_coalesce_max_attempts = 20	(IntOpt) Max number of times to poll for VHD to coalesce. Used only if compute_driver=xenapi.XenAPIDriver
xenserver/vhd_coalesce_poll_interval = 5.0	(FloatOpt) The interval used for polling of coalescing vhds. Used only if compute_driver=xenapi.XenAPIDriver
xenserver/vif_driver = nova.virt.xenapi.vif.XenAPIBridgeDriver	(StrOpt) The XenAPI VIF driver using XenServer Network APIs.

Table 2.63. New default values

Option	Previous default value	New default value
allowed_rpc_exception_modules	nova.exception, cinder.exception, exceptions	oslo.messaging.exceptions, nova.exception, cinder.exception, exceptions
default_log_levels	amqpplib=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, eventlet.wsgi.server=WARN	amqp=WARN, amqpplib=WARN, boto=WARN, qpids=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN
rpc_backend	nova.openstack.common.rpc.impl_kombi	rabbit
rpc_zmq_matchmaker	nova.openstack.common.rpc.matchmaker	oslo.messaging.localhost.MatchMakerLocalhost
running_deleted_instance_action	log	reap
scheduler_default_filters	RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter	RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter
virt_mkfs	default=mkfs.ext3 -L %(fs_label)s -F %(target)s, linux=mkfs.ext3 -L %(fs_label)s -F %(target)s, windows=mkfs.ntfs -force -fast -label %(fs_label)s %(target)s	
baremetal/sql_connection	sqlite://\$state_path/baremetal_ \$sqlite_db	sqlite://\$state_path/ baremetal_nova.sqlite
cells/cell_type	None	compute
database/connection	sqlite:///home/gauvain/sources/ openstack/openstack-doc-tools/ autogenerate_config_docs/sources/ nova/nova/openstack/common/db/ \$sqlite_db	None
database/slave_connection		None
vmware/task_poll_interval	5.0	0.5

Table 2.64. Deprecated options

Deprecated option	New Option
max_kernel_ramdisk_size	xenserver/max_kernel_ramdisk_size
libvirt_type	libvirt/virt_type
xenapi_ipxe_network_name	xenserver/ipxe_network_name
xenapi_sr_base_path	xenserver/sr_base_path
libvirt_cpu_mode	libvirt/cpu_mode

Deprecated option	New Option
xenapi_torrent_seed_chance	xenserver/torrent_seed_chance
xenapi_torrent_max_last_accessed	xenserver/torrent_max_last_accessed
xenapi_connection_username	xenserver/connection_username
libvirt_volume_drivers	libvirt/volume_drivers
libvirt_uri	libvirt/connection_uri
xenapi_check_host	xenserver/check_host
libvirt_inject_password	libvirt/inject_password
xenapi_disable_agent	xenserver/disable_agent
xenapi_connection_password	xenserver/connection_password
xenapi_torrent_max_seeder_processes_per_host	xenserver/torrent_max_seeder_processes_per_host
xenapi_torrent_listen_port_end	xenserver/torrent_listen_port_end
libvirt_wait_soft_reboot_seconds	libvirt/wait_soft_reboot_seconds
libvirt_iscsi_use_multipath	libvirt/iscsi_use_multipath
libvirt_inject_key	libvirt/inject_key
libvirt_inject_partition	libvirt/inject_partition
xenapi_vhd_coalesce_max_attempts	xenserver/vhd_coalesce_max_attempts
xenapi_use_agent_default	xenserver/use_agent_default
libvirt_disk_prefix	libvirt/disk_prefix
libvirt_sparse_logical_volumes	libvirt/sparse_logical_volumes
libvirt_images_volume_group	libvirt/images_volume_group
libvirt_images_rbd_pool	libvirt/images_rbd_pool
xenapi_remap_vbd_dev_prefix	xenserver/remap_vbd_dev_prefix
base_dir_name	image_cache_subdirectory_name
xenapi_torrent_download_stall_cutoff	xenserver/torrent_download_stall_cutoff
xenapi_ipxe_boot_menu_url	xenserver/ipxe_boot_menu_url
sr_matching_filter	xenserver/sr_matching_filter
log_config	log_config_append
libvirt_snapshots_directory	libvirt/snapshots_directory
libvirt_iser_use_multipath	libvirt/iser_use_multipath
libvirt_cpu_model	libvirt/cpu_model
agent_timeout	xenserver/agent_timeout
block_device_creation_timeout	xenserver/block_device_creation_timeout
rpc_notifier2/topics	notification_topics
xenapi_ipxe_mkisofs_cmd	xenserver/ipxe_mkisofs_cmd
xenapi_sparse_copy	xenserver/sparse_copy
use_join_force	xenserver/use_join_force
libvirt_snapshot_compression	libvirt/snapshot_compression
xenapi_num_vbd_unplug_retries	xenserver/num_vbd_unplug_retries
target_port	xenserver/target_port
xenapi_login_timeout	xenserver/login_timeout
xenapi_remap_vbd_dev	xenserver/remap_vbd_dev
xenapi_torrent_listen_port_start	xenserver/torrent_listen_port_start
xenapi_connection_concurrent	xenserver/connection_concurrent
xenapi_vif_driver	xenserver/vif_driver

Deprecated option	New Option
xenapi_torrent_base_url	xenserver/torrent_base_url
xenapi_ovs_integration_bridge	xenserver/ovs_integration_bridge
xenapi_connection_url	xenserver/connection_url
agent_version_timeout	xenserver/agent_version_timeout
xenapi_running_timeout	xenserver/running_timeout
target_host	xenserver/target_host
xenapi_agent_path	xenserver/agent_path
xenapi_vhd_coalesce_poll_interval	xenserver/vhd_coalesce_poll_interval
iqn_prefix	xenserver/iqn_prefix
libvirt_images_type	libvirt/images_type
xenapi_torrent_images	xenserver/torrent_images
xenapi_image_upload_handler	xenserver/image_upload_handler
xenapi_torrent_seed_duration	xenserver/torrent_seed_duration
libvirt_use_virtio_for_bridges	libvirt/use_virtio_for_bridges
cache_images	xenserver/cache_images
libvirt_vif_driver	libvirt/vif_driver
agent_resetnetwork_timeout	xenserver/agent_resetnetwork_timeout
xenapi_image_compression_level	xenserver/image_compression_level
libvirt_images_rbd_ceph_conf	libvirt/images_rbd_ceph_conf
default_os_type	xenserver/default_os_type

3. Dashboard

Table of Contents

Configure the dashboard	253
Customize the dashboard	257
Additional sample configuration files	258
Log files used by the dashboard	269

This chapter describes how to configure the OpenStack dashboard with Apache web server.

Configure the dashboard

You can configure the dashboard for a simple HTTP deployment.

You can configure the dashboard for a secured HTTPS deployment. While the standard installation uses a non-encrypted HTTP channel, you can enable SSL support for the dashboard.

Also, you can configure the size of the VNC window in the dashboard.

Configure the dashboard for HTTP

You can configure the dashboard for a simple HTTP deployment. The standard installation uses a non-encrypted HTTP channel.

1. Specify the host for your OpenStack Identity Service endpoint in the `/etc/openstack-dashboard/local_settings.py` file with the `OPENSTACK_HOST` setting.

The following example shows this setting:

```
import os

from django.utils.translation import ugettext_lazy as _

DEBUG = False
TEMPLATE_DEBUG = DEBUG
PROD = True
USE_SSL = False

SITE_BRANDING = 'OpenStack Dashboard'

# Ubuntu-specific: Enables an extra panel in the 'Settings' section
# that easily generates a Juju environments.yaml for download,
# preconfigured with endpoints and credentials required for bootstrap
# and service deployment.
ENABLE_JUJU_PANEL = True

# Note: You should change this value
SECRET_KEY = 'eljlIWiLoWHgryYxFT6j7cM5fGOOxWY0'

# Specify a regular expression to validate user passwords.
# HORIZON_CONFIG = {
#     "password_validator": {
#         "regex": '.*',
#         "help_text": _("Your password does not meet the requirements.")
#     }
# }
```



```

LOCAL_PATH = os.path.dirname(os.path.abspath(__file__))

CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211'
    }
}

# Send email to the console by default
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
# Or send them to /dev/null
#EMAIL_BACKEND = 'django.core.mail.backends.dummy.EmailBackend'

# Configure these for your outgoing email host
# EMAIL_HOST = 'smtp.my-company.com'
# EMAIL_PORT = 25
# EMAIL_HOST_USER = 'djangomail'
# EMAIL_HOST_PASSWORD = 'top-secret!'

# For multiple regions uncomment this configuration, and add (endpoint, title).
# AVAILABLE_REGIONS = [
#     ('http://cluster1.example.com:5000/v2.0', 'cluster1'),
#     ('http://cluster2.example.com:5000/v2.0', 'cluster2'),
# ]

OPENSTACK_HOST = "127.0.0.1"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "Member"

# The OPENSTACK_KEYSTONE_BACKEND settings can be used to identify the
# capabilities of the auth backend for Keystone.
# If Keystone has been configured to use LDAP as the auth backend then set
# can_edit_user to False and name to 'ldap'.
#
# TODO(tres): Remove these once Keystone has an API to identify auth backend.
OPENSTACK_KEYSTONE_BACKEND = {
    'name': 'native',
    'can_edit_user': True
}

# OPENSTACK_ENDPOINT_TYPE specifies the endpoint type to use for the endpoints
# in the Keystone service catalog. Use this setting when Horizon is running
# external to the OpenStack environment. The default is 'internalURL'.
#OPENSTACK_ENDPOINT_TYPE = "publicURL"

# The number of Swift containers and objects to display on a single page before
# providing a paging element (a "more" link) to paginate results.
API_RESULT_LIMIT = 1000

# If you have external monitoring links, eg:
# EXTERNAL_MONITORING = [
#     ['Nagios', 'http://foo.com'],
#     ['Ganglia', 'http://bar.com'],
# ]

LOGGING = {
    'version': 1,
    # When set to True this will disable all logging except
    # for loggers specified in this configuration dictionary. Note that
    # if nothing is specified here and disable_existing_loggers is True,
    # django.db.backends will still log unless it is disabled explicitly.
    'disable_existing_loggers': False,
    'handlers': {
        'null': {
            'level': 'DEBUG',
            'class': 'django.utils.log.NullHandler',
        },
        'console': {
            # Set the level to "DEBUG" for verbose output logging.
            'level': 'INFO',
            'class': 'logging.StreamHandler',
        },
    },
    'loggers': {
        # Logging from django.db.backends is VERY verbose, send to null
        # by default.
        'django.db.backends': {
            'handlers': ['null'],

```

```

        'propagate': False,
    },
    'horizon': {
        'handlers': ['console'],
        'propagate': False,
    },
    'novaclient': {
        'handlers': ['console'],
        'propagate': False,
    },
    'keystoneclient': {
        'handlers': ['console'],
        'propagate': False,
    },
    'nose.plugins.manager': {
        'handlers': ['console'],
        'propagate': False,
    }
}

```

The service catalog configuration in the Identity Service determines whether a service appears in the dashboard. For the full listing, see [Horizon Settings and Configuration](#).

- Restart Apache http server. For Ubuntu/Debian/SUSE:

```
# service apache2 restart
```

or for Fedora/RHEL/CentOS:

```
# service httpd restart
```

Next, restart memcached:

```
# service memcached restart
```

Configure the dashboard for HTTPS

You can configure the dashboard for a secured HTTPS deployment. While the standard installation uses a non-encrypted HTTP channel, you can enable SSL support for the dashboard.

The following example uses the domain, "http://openstack.example.com." Use a domain that fits your current setup.

- In `/etc/openstack-dashboard/local_settings.py` update the following directives:

```

USE_SSL = True
CSRF_COOKIE_SECURE = True
SESSION_COOKIE_SECURE = True
SESSION_COOKIE_HTTPONLY = True

```

The first option is required to enable HTTPS. The other recommended settings defend against cross-site scripting and require HTTPS.

- Edit `/etc/apache2/ports.conf` and add the following line:

```
NameVirtualHost *:443
```

- Edit `/etc/apache2/conf.d/openstack-dashboard.conf`:

Before:

```

WSGIScriptAlias / /usr/share/openstack-dashboard/openstack_dashboard/wsgi/django.wsgi
WSGIDaemonProcess horizon user=www-data group=www-data processes=3 threads=10
Alias /static /usr/share/openstack-dashboard/openstack_dashboard/static/
<Directory /usr/share/openstack-dashboard/openstack_dashboard/wsgi>
# For Apache http server 2.2 and earlier:
Order allow,deny
Allow from all

# For Apache http server 2.4 and later:
# Require all granted
</Directory>

```

After:

```

<VirtualHost *:80>
ServerName openstack.example.com
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}
</IfModule>
<IfModule !mod_rewrite.c>
RedirectPermanent / https://openstack.example.com
</IfModule>
</VirtualHost>
<VirtualHost *:443>
ServerName openstack.example.com

SSLEngine On
# Remember to replace certificates and keys with valid paths in your environment
SSLCertificateFile /etc/apache2/SSL/openstack.example.com.crt
SSLCACertificateFile /etc/apache2/SSL/openstack.example.com.crt
SSLCertificateKeyFile /etc/apache2/SSL/openstack.example.com.key
SetEnvif User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown

# HTTP Strict Transport Security (HSTS) enforces that all communications
# with a server go over SSL. This mitigates the threat from attacks such
# as SSL-Strip which replaces links on the wire, stripping away https prefixes
# and potentially allowing an attacker to view confidential information on the
# wire
Header add Strict-Transport-Security "max-age=15768000"

WSGIScriptAlias / /usr/share/openstack-dashboard/openstack_dashboard/wsgi/django.wsgi
WSGIDaemonProcess horizon user=www-data group=www-data processes=3 threads=10
Alias /static /usr/share/openstack-dashboard/openstack_dashboard/static/
<Directory /usr/share/openstack-dashboard/openstack_dashboard/wsgi>
# For Apache http server 2.2 and earlier:
Order allow,deny
Allow from all

# For Apache http server 2.4 and later:
# Require all granted
</Directory>
</VirtualHost>

```

In this configuration, Apache http server listens on the port 443 and redirects all the hits to the HTTPS protocol for all the non-secured requests. The secured section defines the private key, public key, and certificate to use.

4. Restart Apache http server. For Debian/Ubuntu/SUSE:

```
# service apache2 restart
```

Or for Fedora/RHEL/CentOS:

```
# service httpd restart
```

Next, restart memcached:

```
# service memcached restart
```

If you try to access the dashboard through HTTP, the browser redirects you to the HTTPS page.

Change the size of the dashboard VNC window

The `_detail_vnc.html` file defines the size of the VNC window. To change the window size, edit this file.

1. Edit `/usr/share/pyshared/horizon/dashboards/nova/instances/templates/instances/_detail_vnc.html`.
2. Modify the `width` and `height` parameters, as follows:

```
<iframe src="{{ vnc_url }}" width="720" height="430"></iframe>
```

Customize the dashboard

Adapted from [How To Custom Brand The OpenStack "Horizon" Dashboard](#).

You install the OpenStack dashboard through the `openstack-dashboard` package. You can customize the dashboard with your own colors, logo, and site title through a CSS file.

Canonical also provides an `openstack-dashboard-ubuntu-theme` package that brands the Python-based Django interface.

1. Create a graphical logo with a transparent background. The text `TGen Cloud` in this example is rendered through `.png` files of multiple sizes created with a graphics program.

Use a `200×27` for the logged-in banner graphic, and `365×50` for the login screen graphic.

2. Set the HTML title, which appears at the top of the browser window, by adding the following line to `/etc/openstack-dashboard/local_settings.py`:

```
SITE_BRANDING = "Example, Inc. Cloud"
```

3. Upload your new graphic files to the following location: `/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/img/`
4. Create a CSS style sheet in the following directory: `/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/css/`
5. Edit your CSS file to override the Ubuntu customizations in the `ubuntu.css` file.

Change the colors and image file names as appropriate, though the relative directory paths should be the same. The following example file shows you how to customize your CSS file:

```
/*  
* New theme colors for dashboard that override the defaults:  
* dark blue: #355796 / rgb(53, 87, 150)  
* light blue: #BAD3E1 / rgb(186, 211, 225)  
*  
* By Preston Lee <plee@tgen.org>
```



```
],
"identity:get_policy": [
  [
    "rule:admin_required"
  ]
],
"identity:list_policies": [
  [
    "rule:admin_required"
  ]
],
"identity:create_policy": [
  [
    "rule:admin_required"
  ]
],
"identity:update_policy": [
  [
    "rule:admin_required"
  ]
],
"identity:delete_policy": [
  [
    "rule:admin_required"
  ]
],
"identity:check_token": [
  [
    "rule:admin_required"
  ]
],
"identity:validate_token": [
  [
    "rule:service_or_admin"
  ]
],
"identity:validate_token_head": [
  [
    "rule:service_or_admin"
  ]
],
"identity:revocation_list": [
  [
    "rule:service_or_admin"
  ]
],
"identity:revoke_token": [
  [
    "rule:admin_or_owner"
  ]
],
"identity:create_trust": [
  [
    "user_id:%(trust.trustor_user_id)s"
  ]
],
"identity:get_trust": [
  [
    "rule:admin_or_owner"
  ]
]
```



```

"compute_extension:v3:os-keypairs:delete": "",
"compute_extension:multinic": "",
"compute_extension:v3:os-multinic": "",
"compute_extension:networks": "rule:admin_api",
"compute_extension:networks:view": "",
"compute_extension:networks_associate": "rule:admin_api",
"compute_extension:quotas:show": "",
"compute_extension:quotas:update": "rule:admin_api",
"compute_extension:quotas:delete": "rule:admin_api",
"compute_extension:v3:os-quota-sets:show": "",
"compute_extension:v3:os-quota-sets:update": "rule:admin_api",
"compute_extension:v3:os-quota-sets:delete": "rule:admin_api",
"compute_extension:quota_classes": "",
"compute_extension:v3:os-quota-class-sets": "",
"compute_extension:rescue": "",
"compute_extension:v3:os-rescue": "",
"compute_extension:security_group_default_rules": "rule:admin_api",
"compute_extension:security_groups": "",
"compute_extension:v3:os-security-groups": "",
"compute_extension:server_diagnostics": "rule:admin_api",
"compute_extension:v3:os-server-diagnostics": "rule:admin_api",
"compute_extension:server_password": "",
"compute_extension:v3:os-server-password": "",
"compute_extension:server_usage": "",
"compute_extension:v3:os-server-usage": "",
"compute_extension:services": "rule:admin_api",
"compute_extension:v3:os-services": "rule:admin_api",
"compute_extension:v3:servers:discoverable": "",
"compute_extension:shelve": "",
"compute_extension:shelveOffload": "rule:admin_api",
"compute_extension:v3:os-shelve:shelve": "",
"compute_extension:v3:os-shelve:shelve_offload": "rule:admin_api",
"compute_extension:simple_tenant_usage:show": "rule:admin_or_owner",
"compute_extension:v3:os-simple-tenant-usage:show": "rule:admin_or_owner",
"compute_extension:simple_tenant_usage:list": "rule:admin_api",
"compute_extension:v3:os-simple-tenant-usage:list": "rule:admin_api",
"compute_extension:unshelve": "",
"compute_extension:v3:os-shelve:unshelve": "",
"compute_extension:users": "rule:admin_api",
"compute_extension:virtual_interfaces": "",
"compute_extension:virtual_storage_arrays": "",
"compute_extension:volumes": "",
"compute_extension:volume_attachments:index": "",
"compute_extension:volume_attachments:show": "",
"compute_extension:volume_attachments:create": "",
"compute_extension:volume_attachments:update": "",
"compute_extension:volume_attachments:delete": "",
"compute_extension:volumetypes": "",
"compute_extension:availability_zone:list": "",
"compute_extension:v3:os-availability-zone:list": "",
"compute_extension:availability_zone:detail": "rule:admin_api",
"compute_extension:v3:os-availability-zone:detail": "rule:admin_api",
"compute_extension:used_limits_for_admin": "rule:admin_api",
"compute_extension:v3:os-used-limits": "",
"compute_extension:v3:os-used-limits:tenant": "rule:admin_api",
"compute_extension:migrations:index": "rule:admin_api",
"compute_extension:v3:os-migrations:index": "rule:admin_api",
"volume:create": "",
"volume:get_all": "",
"volume:get_volume_metadata": "",

```

```

"volume:get_snapshot": "",
"volume:get_all_snapshots": "",
"volume_extension:types_manage": "rule:admin_api",
"volume_extension:types_extra_specs": "rule:admin_api",
"volume_extension:volume_admin_actions:reset_status": "rule:admin_api",
"volume_extension:snapshot_admin_actions:reset_status": "rule:admin_api",
"volume_extension:volume_admin_actions:force_delete": "rule:admin_api",
"network:get_all": "",
"network:get": "",
"network:create": "",
"network:delete": "",
"network:associate": "",
"network:disassociate": "",
"network:get_vifs_by_instance": "",
"network:allocate_for_instance": "",
"network:deallocate_for_instance": "",
"network:validate_networks": "",
"network:get_instance_uuids_by_ip_filter": "",
"network:get_instance_id_by_floating_address": "",
"network:setup_networks_on_host": "",
"network:get_backdoor_port": "",
"network:get_floating_ip": "",
"network:get_floating_ip_pools": "",
"network:get_floating_ip_by_address": "",
"network:get_floating_ips_by_project": "",
"network:get_floating_ips_by_fixed_address": "",
"network:allocate_floating_ip": "",
"network:deallocate_floating_ip": "",
"network:associate_floating_ip": "",
"network:disassociate_floating_ip": "",
"network:release_floating_ip": "",
"network:migrate_instance_start": "",
"network:migrate_instance_finish": "",
"network:get_fixed_ip": "",
"network:get_fixed_ip_by_address": "",
"network:add_fixed_ip_to_instance": "",
"network:remove_fixed_ip_from_instance": "",
"network:add_network_to_project": "",
"network:get_instance_nw_info": "",
"network:get_dns_domains": "",
"network:add_dns_entry": "",
"network:modify_dns_entry": "",
"network:delete_dns_entry": "",
"network:get_dns_entries_by_address": "",
"network:get_dns_entries_by_name": "",
"network:create_private_dns_domain": "",
"network:create_public_dns_domain": "",
"network:delete_dns_domain": ""
}

```

Log files used by the dashboard

The dashboard is served to users through the Apache web server (httpd). As a result, logs relating to the dashboard appear in the following files in the `/var/log/httpd` or `/var/log/apache2` directory of where the dashboard is hosted.

Table 3.1. Log files used by the dashboard/httpd

Log file	Description
access_log	This file logs all attempts to access the web server.
error_log	This file logs all unsuccessful attempts to access the web server, along with the reason each attempt/request failed.

4. Database Service

Table of Contents

Configure the database	278
Configure the RPC messaging system	280

The Database Service provides a scalable and reliable Cloud Database-as-a-Service functionality for both relational and non-relational database engines.

The following tables provide a comprehensive list of the Database Service configuration options.

Table 4.1. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
admin_roles = admin	(ListOpt) No help text available for this option.
api_extensions_path = trove/extensions/routes	(StrOpt) Path to extensions.
api_paste_config = api-paste.ini	(StrOpt) File name for the paste.deploy config for trove-api.
bind_port = 8779	(IntOpt) No help text available for this option.
db_api_implementation = trove.db.sqlalchemy.api	(StrOpt) No help text available for this option.
hostname_require_ipv4 = True	(BoolOpt) Require user hostnames to be IPv4 addresses.
http_delete_rate = 200	(IntOpt) No help text available for this option.
http_get_rate = 200	(IntOpt) No help text available for this option.
http_post_rate = 200	(IntOpt) No help text available for this option.
http_put_rate = 200	(IntOpt) No help text available for this option.
instances_page_size = 20	(IntOpt) No help text available for this option.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs)
region = LOCAL_DEV	(StrOpt) The region this service is located.
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
trove_api_workers = None	(IntOpt) No help text available for this option.
trove_auth_url = http://0.0.0.0:5000/v2.0	(StrOpt) No help text available for this option.
trove_conductor_workers = 1	(IntOpt) No help text available for this option.
trove_security_group_name_prefix = SecGroup	(StrOpt) No help text available for this option.
trove_security_group_rule_cidr = 0.0.0.0/0	(StrOpt) No help text available for this option.
trove_security_groups_support = True	(BoolOpt) No help text available for this option.
users_page_size = 20	(IntOpt) No help text available for this option.

Table 4.2. Description of configuration options for auth_token

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) 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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated

Configuration option = Default value	Description
	in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

Table 4.3. Description of configuration options for backup

Configuration option = Default value	Description
[DEFAULT]	
backup_aes_cbc_key = default_aes_cbc_key	(StrOpt) Default OpenSSL aes_cbc key.
backup_chunk_size = 65536	(IntOpt) Chunk size to stream to swift container. This should be in multiples of 128 bytes, since this is the size of an md5 digest block allowing the process to update the file checksum during streaming. See: http://stackoverflow.com/questions/1131220/
backup_incremental_strategy = {'InnoBackupEx': 'InnoBackupExIncremental'}	(DictOpt) Incremental Backup Runner Based off of the default strategy. For strategies that do not implement an incremental the runner will use the default full backup.
backup_namespace = trove.guestagent.strategies.backup.mysql_impl	(StrOpt) Namespace to load backup strategies from.
backup_runner = trove.guestagent.backup.backup_types.InnoBackupEx	(StrOpt) No help text available for this option.
backup_runner_options = {}	(DictOpt) Additional options to be passed to the backup runner.
backup_segment_max_size = 2147483648	(IntOpt) Maximum size of each segment of the backup file.
backup_strategy = InnoBackupEx	(StrOpt) Default strategy to perform backups.
backup_swift_container = database_backups	(StrOpt) No help text available for this option.
backup_use_gzip_compression = True	(BoolOpt) Compress backups using gzip.
backup_use_openssl_encryption = True	(BoolOpt) Encrypt backups using OpenSSL.
backup_use_snet = False	(BoolOpt) Send backup files over snet.
backups_page_size = 20	(IntOpt) No help text available for this option.

Table 4.4. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
configurations_page_size = 20	(IntOpt) No help text available for this option.
databases_page_size = 20	(IntOpt) No help text available for this option.
default_datastore = None	(StrOpt) The default datastore id or name to use if one is not provided by the user. If the default value is None, the field becomes required in the instance-create request.

Configuration option = Default value	Description
default_log_levels = amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN	(ListOpt) list of logger=LEVEL pairs
default_neutron_networks =	(ListOpt) List of network IDs which should be attached to instance when networks are not specified in API call.
default_notification_level = INFO	(StrOpt) Default notification level for outgoing notifications
default_password_length = 36	(IntOpt) No help text available for this option.
default_publisher_id = \$host	(StrOpt) Default publisher_id for outgoing notifications
expected_filetype_suffixes = json	(ListOpt) Filetype endings not to be reattached to an id by the utils method correct_id_with_req.
lock_path = None	(StrOpt) Directory to use for lock files.
pybasedir = /usr/lib/python/site-packages/trove	(StrOpt) Directory where the trove python module is installed.
pydev_path = None	(StrOpt) Set path to pydevd library, used if pydevd is not found in python sys.path.
taskmanager_queue = taskmanager	(StrOpt) No help text available for this option.
template_path = /etc/trove/templates/	(StrOpt) Path which leads to datastore templates.

Table 4.5. Description of configuration options for compute

Configuration option = Default value	Description
[DEFAULT]	
ip_regex = None	(StrOpt) No help text available for this option.
nova_compute_url = http://localhost:8774/v2	(StrOpt) No help text available for this option.
root_grant = ALL	(ListOpt) No help text available for this option.
root_grant_option = True	(BoolOpt) No help text available for this option.

Table 4.6. Description of configuration options for debug

Configuration option = Default value	Description
[DEFAULT]	
backdoor_port = None	(StrOpt) 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 service's log file.
backlog = 4096	(IntOpt) Number of backlog requests to configure the socket with
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
fatal_deprecations = False	(BoolOpt) make deprecations fatal
pydev_debug = disabled	(StrOpt) Enable or disable pydev remote debugging. If value is 'auto' tries to connect to remote debugger server, but in case of error continues running with debugging disabled.
pydev_debug_host = None	(StrOpt) Pydev debug server host (localhost by default).
pydev_debug_port = None	(IntOpt) Pydev debug server port (5678 by default).

Configuration option = Default value	Description
remote_cinder_client = trove.common.remote.cinder_client	(StrOpt) No help text available for this option.
remote_dns_client = trove.common.remote.dns_client	(StrOpt) No help text available for this option.
remote_guest_client = trove.common.remote.guest_client	(StrOpt) No help text available for this option.
remote_heat_client = trove.common.remote.heat_client	(StrOpt) No help text available for this option.
remote_nova_client = trove.common.remote.nova_client	(StrOpt) No help text available for this option.
remote_swift_client = trove.common.remote.swift_client	(StrOpt) No help text available for this option.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 4.7. Description of configuration options for dns

Configuration option = Default value	Description
[DEFAULT]	
dns_account_id =	(StrOpt) No help text available for this option.
dns_auth_url =	(StrOpt) No help text available for this option.
dns_domain_id =	(StrOpt) No help text available for this option.
dns_domain_name =	(StrOpt) No help text available for this option.
dns_driver = trove.dns.driver.DnsDriver	(StrOpt) No help text available for this option.
dns_endpoint_url = 0.0.0.0	(StrOpt) No help text available for this option.
dns_hostname =	(StrOpt) No help text available for this option.
dns_instance_entry_factory = trove.dns.driver.DnsInstanceEntryFactory	(StrOpt) No help text available for this option.
dns_management_base_url =	(StrOpt) No help text available for this option.
dns_passkey =	(StrOpt) No help text available for this option.
dns_region =	(StrOpt) No help text available for this option.
dns_service_type =	(StrOpt) No help text available for this option.
dns_time_out = 120	(IntOpt) No help text available for this option.
dns_ttl = 300	(IntOpt) No help text available for this option.
dns_username =	(StrOpt) No help text available for this option.
trove_dns_support = False	(BoolOpt) No help text available for this option.

Table 4.8. Description of configuration options for guestagent

Configuration option = Default value	Description
[DEFAULT]	
agent_call_high_timeout = 60	(IntOpt) No help text available for this option.
agent_call_low_timeout = 5	(IntOpt) No help text available for this option.
agent_heartbeat_time = 10	(IntOpt) No help text available for this option.
guest_config = \$pybasedir/etc/trove/trove-guestagent.conf.sample	(StrOpt) Path to guestagent config file.
guest_id = None	(StrOpt) No help text available for this option.
ignore_dbs = lost+found, mysql, information_schema	(ListOpt) No help text available for this option.
ignore_users = os_admin, root	(ListOpt) No help text available for this option.
mount_options = defaults,noatime	(StrOpt) No help text available for this option.
restore_namespace = trove.guestagent.strategies.restore.mysql_impl	(StrOpt) Namespace to load restore strategies from.

Configuration option = Default value	Description
storage_namespace = trove.guestagent.strategies.storage.swift	(StrOpt) Namespace to load the default storage strategy from.
storage_strategy = SwiftStorage	(StrOpt) Default strategy to store backups.
usage_sleep_time = 5	(IntOpt) Time to sleep during the check active guest.

Table 4.9. Description of configuration options for heat

Configuration option = Default value	Description
[DEFAULT]	
heat_time_out = 60	(IntOpt) No help text available for this option.
heat_url = http://localhost:8004/v1	(StrOpt) No help text available for this option.

Table 4.10. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
format_options = -m 5	(StrOpt) No help text available for this option.
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s. %(msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s %(user_identity)s] %(instance)s %(message)s	(StrOpt) format string to use for log messages with context
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s. %(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) format string to use for log messages without context
logging_exception_prefix = %(asctime)s. %(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) prefix each line of exception output with this format
network_label_regex = ^private\$	(StrOpt) No help text available for this option.
publish_errors = False	(BoolOpt) publish error events
syslog_log_facility = LOG_USER	(StrOpt) syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging.

Table 4.11. Description of configuration options for nova

Configuration option = Default value	Description
[DEFAULT]	
nova_proxy_admin_pass =	(StrOpt) Admin password used to connect to nova,
nova_proxy_admin_tenant_name =	(StrOpt) Admin tenant used to connect to nova.
nova_proxy_admin_user =	(StrOpt) Admin username used to connect to nova.

Table 4.12. Description of configuration options for quota

Configuration option = Default value	Description
[DEFAULT]	
max_accepted_volume_size = 5	(IntOpt) Default maximum volume size for an instance.
max_backups_per_user = 50	(IntOpt) Default maximum number of backups created by a tenant.
max_instances_per_user = 5	(IntOpt) Default maximum number of instances per tenant.
max_volumes_per_user = 20	(IntOpt) Default maximum volume capacity (in GB) spanning across all trove volumes per tenant
quota_driver = trove.quota.quota.DbQuotaDriver	(StrOpt) Default driver to use for quota checks.

Table 4.13. Description of configuration options for redis

Configuration option = Default value	Description
[matchmaker_redis]	
host = 127.0.0.1	(StrOpt) Host to locate redis
password = None	(StrOpt) Password for Redis server. (optional)
port = 6379	(IntOpt) Use this port to connect to redis host.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON)

Table 4.14. Description of configuration options for ssl

Configuration option = Default value	Description
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 4.15. Description of configuration options for swift

Configuration option = Default value	Description
[DEFAULT]	
swift_url = http://localhost:8080/v1/AUTH_	(StrOpt) No help text available for this option.

Table 4.16. Description of configuration options for taskmanager

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
cloudinit_location = /etc/trove/cloudinit	(StrOpt) Path to folder with cloudinit scripts.
datastore_manager = None	(StrOpt) Manager class in guestagent, setup by taskmanager on instance provision.
datastore_registry_ext = {}	(DictOpt) Extension for default datastore managers. Allows to use custom managers for each of datastore supported in trove.
exists_notification_ticks = 360	(IntOpt) Number of report_intervals to wait between pushing events (see report_interval).
exists_notification_transformer = None	(StrOpt) Transformer for exists notifications.
reboot_time_out = 120	(IntOpt) No help text available for this option.
resize_time_out = 600	(IntOpt) No help text available for this option.
revert_time_out = 600	(IntOpt) No help text available for this option.
server_delete_time_out = 60	(IntOpt) No help text available for this option.
state_change_wait_time = 180	(IntOpt) No help text available for this option.
update_status_on_fail = False	(BoolOpt) If instance fails to become active, taskmanager updates statuses, service status = FAILED_TIMEOUT_GUESTAGENT, instance task status = BUILDING_ERROR_TIMEOUT_GA.
usage_sleep_time = 5	(IntOpt) Time to sleep during the check active guest.
use_heat = False	(BoolOpt) No help text available for this option.
use_nova_server_volume = False	(BoolOpt) No help text available for this option.
verify_swift_checksum_on_restore = True	(BoolOpt) Enable verification of swift checksum before starting restore; makes sure the checksum of original backup matches checksum of the swift backup file.

Table 4.17. Description of configuration options for volume

Configuration option = Default value	Description
[DEFAULT]	
block_device_mapping = vdb	(StrOpt) No help text available for this option.
cinder_url = http://localhost:8776/v2	(StrOpt) No help text available for this option.
device_path = /dev/vdb	(StrOpt) No help text available for this option.
trove_volume_support = True	(BoolOpt) Whether to provision a cinder volume for datadir.
volume_format_timeout = 120	(IntOpt) No help text available for this option.
volume_fstype = ext3	(StrOpt) No help text available for this option.
volume_time_out = 60	(IntOpt) No help text available for this option.

Configure the database

Use the options to configure the used databases:

Table 4.18. Description of configuration options for database

Configuration option = Default value	Description
[DEFAULT]	
sql_connection = sqlite:///trove_test.sqlite	(StrOpt) SQL Connection.
sql_idle_timeout = 3600	(IntOpt) No help text available for this option.
sql_query_log = False	(BoolOpt) No help text available for this option.

Configuration option = Default value	Description
sql_query_logging = False	(BoolOpt) Allow insecure logging while executing queries through SQLAlchemy.

Table 4.19. Description of configuration options for db_cassandra

Configuration option = Default value	Description
[cassandra]	
backup_strategy = None	(StrOpt) Default strategy to perform backups.
mount_point = /var/lib/cassandra	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
tcp_ports = 7000, 7001, 9042, 9160	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 600	(IntOpt) Timeout to wait for a guest to become active.

Table 4.20. Description of configuration options for db_couchbase

Configuration option = Default value	Description
[couchbase]	
backup_strategy = None	(StrOpt) Default strategy to perform backups.
mount_point = /var/lib/couchbase	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
tcp_ports = 8091, 8092, 4369, 11209-11211, 21100-21199	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 450	(IntOpt) Timeout to wait for a guest to become active.

Table 4.21. Description of configuration options for db_mongodb

Configuration option = Default value	Description
[mongodb]	
backup_strategy = None	(StrOpt) Default strategy to perform backups.
mount_point = /var/lib/mongodb	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
tcp_ports = 2500, 27017	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UPD ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 450	(IntOpt) Timeout to wait for a guest to become active.

Table 4.22. Description of configuration options for db_mysql

Configuration option = Default value	Description
[mysql]	
backup_strategy = InnoDBBackupEx	(StrOpt) Default strategy to perform backups.

Configuration option = Default value	Description
mount_point = /var/lib/mysql	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
root_on_create = False	(BoolOpt) Enable the automatic creation of the root user for the service during instance-create. The generated password for the root user is immediately returned in the response of instance-create as the 'password' field.
tcp_ports = 3306	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 400	(IntOpt) Timeout to wait for a guest to become active.

Table 4.23. Description of configuration options for db_percona

Configuration option = Default value	Description
[percona]	
backup_strategy = InnoDBBackupEx	(StrOpt) Default strategy to perform backups.
mount_point = /var/lib/mysql	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
root_on_create = False	(BoolOpt) Enable the automatic creation of the root user for the service during instance-create. The generated password for the root user is immediately returned in the response of instance-create as the 'password' field.
tcp_ports = 3306	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 450	(IntOpt) Timeout to wait for a guest to become active.

Table 4.24. Description of configuration options for db_redis

Configuration option = Default value	Description
[redis]	
backup_strategy = None	(StrOpt) Default strategy to perform backups.
mount_point = /var/lib/redis	(StrOpt) Filesystem path for mounting volumes if volume support is enabled
tcp_ports = 6379	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
udp_ports =	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True)
usage_timeout = 450	(IntOpt) Timeout to wait for a guest to become active.

Configure the RPC messaging system

OpenStack projects use an open standard for messaging middleware known as AMQP. This messaging middleware enables the OpenStack services that run on multiple servers to talk to each other. OpenStack Trove RPC supports three implementations of AMQP: RabbitMQ, Qpid, and ZeroMQ.

Configure RabbitMQ

Use these options to configure the RabbitMQ messaging system:

Table 4.25. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
rabbit_ha_queues = False	(BoolOpt) use H/A queues in RabbitMQ (x-ha-policy: all). You need to wipe RabbitMQ database when changing this option.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs
rabbit_max_retries = 0	(IntOpt) maximum retries with trying to connect to RabbitMQ (the default of 0 implies an infinite retry count)
rabbit_password = guest	(StrOpt) the RabbitMQ password
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used
rabbit_retry_backoff = 2	(IntOpt) how long to backoff for between retries when connecting to RabbitMQ
rabbit_retry_interval = 1	(IntOpt) how frequently to retry connecting with RabbitMQ
rabbit_use_ssl = False	(BoolOpt) connect over SSL for RabbitMQ
rabbit_userid = guest	(StrOpt) the RabbitMQ userid
rabbit_virtual_host = /	(StrOpt) the RabbitMQ virtual host

Configure Qpid

Use these options to configure the Qpid messaging system:

Table 4.26. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats
qpid_hostname = localhost	(StrOpt) Qpid broker hostname
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs
qpid_password =	(StrOpt) Password for qpid connection
qpid_port = 5672	(IntOpt) Qpid broker port
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth
qpid_tcp_nodelay = True	(BoolOpt) Disable Nagle algorithm
qpid_username =	(StrOpt) Username for qpid connection

Configure ZeroMq

Use these options to configure the ZeroMq messaging system:

Table 4.27. Description of configuration options for zeromq

Configuration option = Default value	Description
[DEFAULT]	
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets
rpc_zmq_matchmaker = trove.openstack.common.rpc.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Configure messaging

Use these common options to configure the RabbitMQ, Qpid, and ZeroMq messaging drivers:

Table 4.28. Description of configuration options for amqp

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
conductor_queue = trove-conductor	(StrOpt) No help text available for this option.
control_exchange = openstack	(StrOpt) AMQP exchange to connect to if using RabbitMQ or Qpid
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled)
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled)
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled)
kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications
notification_service_id = {'couchbase': 'fa62fe68-74d9-4779-a24e-36f19602c415', 'mongodb': 'c8c907af-7375-456f-b929-b637ff9209ee', 'cassandra': '459a230d-4e97-4344-9067-2a54a310b0ed', 'redis': 'b216ffc5-1947-456c-a4cf-70f94c05f7d0', 'mysql': '2f3ff068-2bfb-4f70-9a9d-a6bb65bc084b'}	(DictOpt) Unique ID to tag notification events.
notification_topics = notifications	(ListOpt) AMQP topic used for openstack notifications

Table 4.29. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
allowed_rpc_exception_modules = nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.
host = 0.0.0.0	(StrOpt) No help text available for this option.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
num_tries = 3	(IntOpt) No help text available for this option.
periodic_interval = 60	(IntOpt) No help text available for this option.
report_interval = 10	(IntOpt) The interval in seconds which periodic tasks are run.
rpc_backend = trove.openstack.common.rpc.impl_kombu	(StrOpt) The messaging module to use, defaults to kombu.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from call or multical
rpc_thread_pool_size = 64	(IntOpt) Size of RPC thread pool
[rpc_notifier2]	
topics = notifications	(ListOpt) AMQP topic(s) used for openstack notifications
[secure_messages]	
enabled = True	(BoolOpt) Whether Secure Messaging (Signing) is enabled, defaults to enabled
encrypt = False	(BoolOpt) Whether Secure Messaging (Encryption) is enabled, defaults to not enabled
enforced = False	(BoolOpt) Whether Secure Messaging (Signing) is enforced, defaults to not enforced
kds_endpoint = None	(StrOpt) KDS endpoint (ex: http://kds.example.com:35357/v3)
secret_key = None	(MultiStrOpt) A list of keys: (ex: name:<base64 encoded key>), ignored if secret_keys_file is set
secret_keys_file = None	(StrOpt) Path to the file containing the keys, takes precedence over secret_key

5. Identity service

Table of Contents

Identity service configuration file	284
Identity service sample configuration files	298
New, updated and deprecated options for keystone	326

This chapter details the OpenStack Identity service configuration options. For installation prerequisites and step-by-step walkthroughs, see the *OpenStack Installation Guide* for your distribution (docs.openstack.org) and [Cloud Administrator Guide](#).

Identity service configuration file

The Identity service is configured in the `/etc/keystone/keystone.conf` file.

The following tables provide a comprehensive list of the Identity service options.

Table 5.1. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
admin_bind_host = 0.0.0.0	(StrOpt) The IP Address of the network interface to for the admin service to listen on.
admin_endpoint = None	(StrOpt) The base admin endpoint URL for keystone that are advertised to clients (NOTE: this does NOT affect how keystone listens for connections). Defaults to the base host URL of the request. Eg a request to <code>http://server:35357/v2.0/users</code> will default to <code>http://server:35357</code> . You should only need to set this value if the base URL contains a path (eg <code>/prefix/v2.0</code>) or the endpoint should be found on a different server.
admin_port = 35357	(IntOpt) The port number which the admin service listens on.
admin_token = ADMIN	(StrOpt) A "shared secret" that can be used to bootstrap Keystone. This "token" does not represent a user, and carries no explicit authorization. To disable in production (highly recommended), remove <code>AdminTokenAuthMiddleware</code> from your paste application pipelines (for example, in <code>keystone-paste.ini</code>).
compute_port = 8774	(IntOpt) The port which the OpenStack Compute service listens on.
domain_id_immutable = True	(BoolOpt) Set this to false if you want to enable the ability for user, group and project entities to be moved between domains by updating their <code>domain_id</code> . Allowing such movement is not recommended if the scope of a domain admin is being restricted by use of an appropriate policy file (see <code>policy.v3cloudsample</code> as an example).
list_limit = None	(IntOpt) The maximum number of entities that will be returned in a collection can be set with <code>list_limit</code> , with no limit set by default. This global limit may be then

Configuration option = Default value	Description
	overridden for a specific driver, by specifying a list_limit in the appropriate section (e.g. [assignment]).
max_param_size = 64	(IntOpt) limit the sizes of user & tenant ID/names.
max_request_body_size = 114688	(IntOpt) enforced by optional sizelimit middleware (keystone.middleware:RequestBodySizeLimiter).
max_token_size = 8192	(IntOpt) similar to max_param_size, but provides an exception for token values.
member_role_id = 9fe2ff9ee4384b1894a90878d3e92bab	(StrOpt) During a SQL upgrade member_role_id will be used to create a new role that will replace records in the user_tenant_membership table with explicit role grants. After migration, the member_role_id will be used in the API add_user_to_project.
member_role_name = _member_	(StrOpt) During a SQL upgrade member_role_id will be used to create a new role that will replace records in the user_tenant_membership table with explicit role grants. After migration, member_role_name will be ignored.
public_bind_host = 0.0.0.0	(StrOpt) The IP Address of the network interface to for the public service to listen on.
public_endpoint = None	(StrOpt) The base public endpoint URL for keystone that are advertised to clients (NOTE: this does NOT affect how keystone listens for connections). Defaults to the base host URL of the request. Eg a request to http://server:5000/v2.0/users will default to http://server:5000. You should only need to set this value if the base URL contains a path (eg /prefix/v2.0) or the endpoint should be found on a different server.
public_port = 5000	(IntOpt) The port number which the public service listens on.
tcp_keepalive = False	(BoolOpt) Set this to True if you want to enable TCP_KEEPALIVE on server sockets i.e. sockets used by the keystone wsgi server for client connections.
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Only applies if tcp_keepalive is True. Not supported on OS X.
[endpoint_filter]	
driver = keystone.contrib.endpoint_filter.backends.sql.EndpointFilter	(StrOpt) Keystone Endpoint Filter backend driver
return_all_endpoints_if_no_filter = True	(BoolOpt) Toggle to return all active endpoints if no filter exists.
[paste_deploy]	
config_file = keystone-paste.ini	(StrOpt) Name of the paste configuration file that defines the available pipelines.

Table 5.2. Description of configuration options for assignment

Configuration option = Default value	Description
[assignment]	
cache_time = None	(IntOpt) TTL (in seconds) to cache assignment data. This has no effect unless global caching is enabled.
caching = True	(BoolOpt) Toggle for assignment caching. This has no effect unless global caching is enabled.
driver = None	(StrOpt) Keystone Assignment backend driver.
list_limit = None	(IntOpt) Maximum number of entities that will be returned in an assignment collection.

Table 5.3. Description of configuration options for auth

Configuration option = Default value	Description
[auth]	
external = keystone.auth.plugins.external.DefaultDomain	(StrOpt) The external (REMOTE_USER) auth plugin module.
methods = external, password, token	(ListOpt) Default auth methods.
password = keystone.auth.plugins.password.Password	(StrOpt) The password auth plugin module.
token = keystone.auth.plugins.token.Token	(StrOpt) The token auth plugin module.

Table 5.4. Description of configuration options for cache

Configuration option = Default value	Description
[cache]	
backend = keystone.common.cache.noop	(StrOpt) Dogpile.cache backend module. It is recommended that Memcache (dogpile.cache.memcache) or Redis (dogpile.cache.redis) be used in production deployments. Small workloads (single process) like devstack can use the dogpile.cache.memory backend.
backend_argument = []	(MultiStrOpt) Arguments supplied to the backend module. Specify this option once per argument to be passed to the dogpile.cache backend. Example format: "<argname>:<value>".
config_prefix = cache.keystone	(StrOpt) Prefix for building the configuration dictionary for the cache region. This should not need to be changed unless there is another dogpile.cache region with the same configuration name.
debug_cache_backend = False	(BoolOpt) Extra debugging from the cache backend (cache keys, get/set/delete/etc calls) This is only really useful if you need to see the specific cache-backend get/set/delete calls with the keys/values. Typically this should be left set to False.
enabled = False	(BoolOpt) Global toggle for all caching using the should_cache_fn mechanism.
expiration_time = 600	(IntOpt) Default TTL, in seconds, for any cached item in the dogpile.cache region. This applies to any cached method that doesn't have an explicit cache expiration time defined for it.
proxies =	(ListOpt) Proxy Classes to import that will affect the way the dogpile.cache backend functions. See the dogpile.cache documentation on changing-backend-behavior. Comma delimited list e.g. my.dogpile.proxy.Class, my.dogpile.proxyClass2.
use_key_mangler = True	(BoolOpt) Use a key-mangling function (sha1) to ensure fixed length cache-keys. This is toggle-able for debugging purposes, it is highly recommended to always leave this set to True.

Table 5.5. Description of configuration options for catalog

Configuration option = Default value	Description
[catalog]	
driver = keystone.catalog.backends.sql.Catalog	(StrOpt) Keystone catalog backend driver.
list_limit = None	(IntOpt) Maximum number of entities that will be returned in a catalog collection.
template_file = default_catalog.templates	(StrOpt) Catalog template file name for use with the template catalog backend.

Table 5.6. Description of configuration options for credential

Configuration option = Default value	Description
[credential]	
driver = keystone.credential.backends.sql.Credential	(StrOpt) Keystone Credential backend driver.

Table 5.7. Description of configuration options for database

Configuration option = Default value	Description
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
connection = None	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
db_retry_interval = 1	(IntOpt) seconds between db connection retries
idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
mysql_sql_mode = TRADITIONAL	(StrOpt) 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=
pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
sqlite_db = keystone.sqlite	(StrOpt) The file name to use with SQLite
sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost

Table 5.8. Description of configuration options for debug

Configuration option = Default value	Description
[DEFAULT]	
backdoor_port = None	(StrOpt) 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

Configuration option = Default value	Description
	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 service's log file.
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
fatal_deprecations = False	(BoolOpt) Make deprecations fatal
publish_errors = False	(BoolOpt) Publish error events
pydev_debug_host = None	(StrOpt) Host to connect to for remote debugger.
pydev_debug_port = None	(IntOpt) Port to connect to for remote debugger.
standard_threads = False	(BoolOpt) Do not monkey-patch threading system modules.
[audit]	
namespace = openstack	(StrOpt) namespace prefix for generated id

Table 5.9. Description of configuration options for ec2

Configuration option = Default value	Description
[DEFAULT]	
keystone_ec2_cafile = None	(StrOpt) A PEM encoded certificate authority to use when verifying HTTPS connections. Defaults to the system CAs.
keystone_ec2_certfile = None	(StrOpt) Client certificate key filename. Required if EC2 server requires client certificate.
keystone_ec2_insecure = False	(BoolOpt) Disable SSL certificate verification.
keystone_ec2_keyfile = None	(StrOpt) Required if EC2 server requires client certificate.
keystone_ec2_url = http://localhost:5000/v2.0/ec2tokens	(StrOpt) URL to get token from ec2 request.
[ec2]	
driver = keystone.contrib.ec2.backends.kvs.Ec2	(StrOpt) Keystone EC2Credential backend driver.

Table 5.10. Description of configuration options for federation

Configuration option = Default value	Description
[federation]	
assertion_prefix =	(StrOpt) Value to be used when filtering assertion parameters from the environment.
driver = keystone.contrib.federation.backends.sql.Federation	(StrOpt) Keystone Federation backend driver.

Table 5.11. Description of configuration options for identity

Configuration option = Default value	Description
[identity]	
default_domain_id = default	(StrOpt) This references the domain to use for all Identity API v2 requests (which are not aware of domains). A domain with this ID will be created for you by keystone-manage db_sync in migration 008. The domain referenced by this ID cannot be deleted on the v3 API, to prevent accidentally breaking the v2 API. There is nothing special about this domain, other than the fact that it must exist to order to maintain support for your v2 clients.

Configuration option = Default value	Description
domain_config_dir = /etc/keystone/domains	(StrOpt) Path for Keystone to locate the domain specific identity configuration files if domain_specific_drivers_enabled is set to true.
domain_specific_drivers_enabled = False	(BoolOpt) A subset (or all) of domains can have their own identity driver, each with their own partial configuration file in a domain configuration directory. Only values specific to the domain need to be placed in the domain specific configuration file. This feature is disabled by default; set to True to enable.
driver = keystone.identity.backends.sql.Identity	(StrOpt) Keystone Identity backend driver.
list_limit = None	(IntOpt) Maximum number of entities that will be returned in an identity collection.
max_password_length = 4096	(IntOpt) Maximum supported length for user passwords; decrease to improve performance.

Table 5.12. Description of configuration options for kvs

Configuration option = Default value	Description
[kvs]	
backends =	(ListOpt) Extra dogpile.cache backend modules to register with the dogpile.cache library.
config_prefix = keystone.kvs	(StrOpt) Prefix for building the configuration dictionary for the KVS region. This should not need to be changed unless there is another dogpile.cache region with the same configuration name.
default_lock_timeout = 5	(IntOpt) Default lock timeout for distributed locking.
enable_key_mangler = True	(BoolOpt) Toggle to disable using a key-mangling function to ensure fixed length keys. This is toggle-able for debugging purposes, it is highly recommended to always leave this set to True.

Table 5.13. Description of configuration options for ldap

Configuration option = Default value	Description
[ldap]	
alias_dereferencing = default	(StrOpt) The LDAP dereferencing option for queries. This can be either "never", "searching", "always", "finding" or "default". The "default" option falls back to using default dereferencing configured by your ldap.conf.
allow_subtree_delete = False	(BoolOpt) allow deleting subtrees.
chase_referrals = None	(BoolOpt) Override the system's default referral chasing behavior for queries.
dumb_member = cn=dumb,dc=nonexistent	(StrOpt) DN of the "dummy member" to use when "use_dumb_member" is enabled.
group_additional_attribute_mapping =	(ListOpt) Additional attribute mappings for groups. Attribute mapping format is <ldap_attr>:<user_attr>, where ldap_attr is the attribute in the LDAP entry and user_attr is the Identity API attribute.
group_allow_create = True	(BoolOpt) Allow group creation in LDAP backend.
group_allow_delete = True	(BoolOpt) Allow group deletion in LDAP backend.
group_allow_update = True	(BoolOpt) Allow group update in LDAP backend.
group_attribute_ignore =	(ListOpt) List of attributes stripped off the group on update.
group_desc_attribute = description	(StrOpt) LDAP attribute mapped to group description.

Configuration option = Default value	Description
group_filter = None	(StrOpt) LDAP search filter for groups.
group_id_attribute = cn	(StrOpt) LDAP attribute mapped to group id.
group_member_attribute = member	(StrOpt) LDAP attribute mapped to show group membership.
group_name_attribute = ou	(StrOpt) LDAP attribute mapped to group name.
group_objectclass = groupOfNames	(StrOpt) LDAP objectClass for groups.
group_tree_dn = None	(StrOpt) Search base for groups.
page_size = 0	(IntOpt) Maximum results per page; a value of zero ("0") disables paging.
password = None	(StrOpt) Password for the BindDN to query the LDAP server.
query_scope = one	(StrOpt) The LDAP scope for queries, this can be either "one" (onelevel/singleLevel) or "sub" (subtree/wholeSubtree).
role_additional_attribute_mapping =	(ListOpt) Additional attribute mappings for roles. Attribute mapping format is <ldap_attr>:<user_attr>, where ldap_attr is the attribute in the LDAP entry and user_attr is the Identity API attribute.
role_allow_create = True	(BoolOpt) Allow role creation in LDAP backend.
role_allow_delete = True	(BoolOpt) Allow role deletion in LDAP backend.
role_allow_update = True	(BoolOpt) Allow role update in LDAP backend.
role_attribute_ignore =	(ListOpt) List of attributes stripped off the role on update.
role_filter = None	(StrOpt) LDAP search filter for roles.
role_id_attribute = cn	(StrOpt) LDAP attribute mapped to role id.
role_member_attribute = roleOccupant	(StrOpt) LDAP attribute mapped to role membership.
role_name_attribute = ou	(StrOpt) LDAP attribute mapped to role name.
role_objectclass = organizationalRole	(StrOpt) LDAP objectClass for roles.
role_tree_dn = None	(StrOpt) Search base for roles.
suffix = cn=example,cn=com	(StrOpt) LDAP server suffix
tenant_additional_attribute_mapping =	(ListOpt) Additional attribute mappings for projects. Attribute mapping format is <ldap_attr>:<user_attr>, where ldap_attr is the attribute in the LDAP entry and user_attr is the Identity API attribute.
tenant_allow_create = True	(BoolOpt) Allow tenant creation in LDAP backend.
tenant_allow_delete = True	(BoolOpt) Allow tenant deletion in LDAP backend.
tenant_allow_update = True	(BoolOpt) Allow tenant update in LDAP backend.
tenant_attribute_ignore =	(ListOpt) List of attributes stripped off the project on update.
tenant_desc_attribute = description	(StrOpt) LDAP attribute mapped to project description.
tenant_domain_id_attribute = businessCategory	(StrOpt) LDAP attribute mapped to project domain_id.
tenant_enabled_attribute = enabled	(StrOpt) LDAP attribute mapped to project enabled.
tenant_enabled_emulation = False	(BoolOpt) If True, Keystone uses an alternative method to determine if a project is enabled or not by checking if they are a member of the "tenant_enabled_emulation_dn" group.
tenant_enabled_emulation_dn = None	(StrOpt) DN of the group entry to hold enabled projects when using enabled emulation.
tenant_filter = None	(StrOpt) LDAP search filter for projects.
tenant_id_attribute = cn	(StrOpt) LDAP attribute mapped to project id.

Configuration option = Default value	Description
tenant_member_attribute = member	(StrOpt) LDAP attribute mapped to project membership for user.
tenant_name_attribute = ou	(StrOpt) LDAP attribute mapped to project name.
tenant_objectclass = groupOfNames	(StrOpt) LDAP objectClass for projects.
tenant_tree_dn = None	(StrOpt) Search base for projects
tls_cacertdir = None	(StrOpt) CA certificate directory path for communicating with LDAP servers.
tls_cacertfile = None	(StrOpt) CA certificate file path for communicating with LDAP servers.
tls_req_cert = demand	(StrOpt) valid options for tls_req_cert are demand, never, and allow.
url = ldap://localhost	(StrOpt) URL for connecting to the LDAP server.
use_dumb_member = False	(BoolOpt) If true, will add a dummy member to groups. This is required if the objectclass for groups requires the "member" attribute.
use_tls = False	(BoolOpt) Enable TLS for communicating with LDAP servers.
user = None	(StrOpt) User BindDN to query the LDAP server.
user_additional_attribute_mapping =	(ListOpt) List of additional LDAP attributes used for mapping Additional attribute mappings for users. Attribute mapping format is <ldap_attr>:<user_attr>, where ldap_attr is the attribute in the LDAP entry and user_attr is the Identity API attribute.
user_allow_create = True	(BoolOpt) Allow user creation in LDAP backend.
user_allow_delete = True	(BoolOpt) Allow user deletion in LDAP backend.
user_allow_update = True	(BoolOpt) Allow user updates in LDAP backend.
user_attribute_ignore = default_project_id, tenants	(ListOpt) List of attributes stripped off the user on update.
user_default_project_id_attribute = None	(StrOpt) LDAP attribute mapped to default_project_id for users.
user_enabled_attribute = enabled	(StrOpt) LDAP attribute mapped to user enabled flag.
user_enabled_default = True	(StrOpt) Default value to enable users. This should match an appropriate int value if the LDAP server uses non-boolean (bitmask) values to indicate if a user is enabled or disabled. If this is not set to "True" the typical value is "512". This is typically used when "user_enabled_attribute = userAccountControl".
user_enabled_emulation = False	(BoolOpt) If True, Keystone uses an alternative method to determine if a user is enabled or not by checking if they are a member of the "user_enabled_emulation_dn" group.
user_enabled_emulation_dn = None	(StrOpt) DN of the group entry to hold enabled users when using enabled emulation.
user_enabled_mask = 0	(IntOpt) Bitmask integer to indicate the bit that the enabled value is stored in if the LDAP server represents "enabled" as a bit on an integer rather than a boolean. A value of "0" indicates the mask is not used. If this is not set to "0" the typical value is "2". This is typically used when "user_enabled_attribute = userAccountControl".
user_filter = None	(StrOpt) LDAP search filter for users.
user_id_attribute = cn	(StrOpt) LDAP attribute mapped to user id.
user_mail_attribute = email	(StrOpt) LDAP attribute mapped to user email.
user_name_attribute = sn	(StrOpt) LDAP attribute mapped to user name.
user_objectclass = inetOrgPerson	(StrOpt) LDAP objectClass for users.

Configuration option = Default value	Description
user_pass_attribute = userPassword	(StrOpt) LDAP attribute mapped to password.
user_tree_dn = None	(StrOpt) Search base for users.

Table 5.14. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
default_log_levels = amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN	(ListOpt) List of logger=LEVEL pairs
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s. %(msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s %(user_identity)s] %(instance)s %(message)s	(StrOpt) Format string to use for log messages with context
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) Data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) Format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) Prefix each line of exception output with this format
syslog_log_facility = LOG_USER	(StrOpt) Syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and then will be changed in J to honor RFC5424
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 5.15. Description of configuration options for memcache

Configuration option = Default value	Description
[memcache]	
max_compare_and_set_retry = 16	(IntOpt) Number of compare-and-set attempts to make when using compare-and-set in the token memcache back end.
servers = localhost:11211	(ListOpt) Memcache servers in the format of "host:port"

Table 5.16. Description of configuration options for misc

Configuration option = Default value	Description
[DEFAULT]	
lock_path = None	(StrOpt) Directory to use for lock files.

Table 5.17. Description of configuration options for notification

Configuration option = Default value	Description
[DEFAULT]	
onready = None	(StrOpt) onready allows you to send a notification when the process is ready to serve For example, to have it notify using systemd, one could set shell command: "onready = systemd-notify --ready" or a module with notify() method: "onready = keystone.common.systemd".

Table 5.18. Description of configuration options for oauth

Configuration option = Default value	Description
[oauth1]	
access_token_duration = 86400	(IntOpt) Duration (in seconds) for the OAuth Access Token.
driver = keystone.contrib.oauth1.backends.sql.OAuth1	(StrOpt) Keystone Credential backend driver.
request_token_duration = 28800	(IntOpt) Duration (in seconds) for the OAuth Request Token.

Table 5.19. Description of configuration options for os_inherit

Configuration option = Default value	Description
[os_inherit]	
enabled = False	(BoolOpt) role-assignment inheritance to projects from owning domain can be optionally enabled.

Table 5.20. Description of configuration options for policy

Configuration option = Default value	Description
[DEFAULT]	
policy_default_rule = default	(StrOpt) Rule enforced when requested rule is not found
policy_file = policy.json	(StrOpt) JSON file containing policy
[policy]	
driver = keystone.policy.backends.sql.Policy	(StrOpt) Keystone Policy backend driver.
list_limit = None	(IntOpt) Maximum number of entities that will be returned in a policy collection.

Table 5.21. Description of configuration options for revoke

Configuration option = Default value	Description
[revoke]	
cacheing = True	(BoolOpt) Toggle for revocation event cacheing. This has no effect unless global caching is enabled.
driver = keystone.contrib.revoke.backends.kvs.Revoke	(StrOpt) An implementation of the backend for persisting revocation events.
expiration_buffer = 1800	(IntOpt) This value (calculated in seconds) is added to token expiration before a revocation event may be removed from the backend.

Table 5.22. Description of configuration options for security

Configuration option = Default value	Description
[DEFAULT]	
crypt_strength = 40000	(IntOpt) The value passed as the keyword "rounds" to passlib encrypt method.

Table 5.23. Description of configuration options for ssl

Configuration option = Default value	Description
[signing]	
ca_certs = /etc/keystone/ssl/certs/ca.pem	(StrOpt) Path of the CA for token signing.
ca_key = /etc/keystone/ssl/private/cakey.pem	(StrOpt) Path of the CA Key for token signing.
cert_subject = /C=US/ST=Unset/L=Unset/O=Unset/CN=www.example.com	(StrOpt) Certificate Subject (auto generated certificate) for token signing.
certfile = /etc/keystone/ssl/certs/signing_cert.pem	(StrOpt) Path of the certfile for token signing.
key_size = 2048	(IntOpt) Key Size (in bits) for token signing cert (auto generated certificate).
keyfile = /etc/keystone/ssl/private/signing_key.pem	(StrOpt) Path of the keyfile for token signing.
token_format = None	(StrOpt) Deprecated in favor of provider in the [token] section.
valid_days = 3650	(IntOpt) Day the token signing cert is valid for (auto generated certificate).
[ssl]	
ca_certs = /etc/keystone/ssl/certs/ca.pem	(StrOpt) Path of the ca cert file for SSL.
ca_key = /etc/keystone/ssl/private/cakey.pem	(StrOpt) Path of the CA key file for SSL.
cert_required = False	(BoolOpt) Require client certificate.
cert_subject = /C=US/ST=Unset/L=Unset/O=Unset/CN=localhost	(StrOpt) SSL Certificate Subject (auto generated certificate).
certfile = /etc/keystone/ssl/certs/keystone.pem	(StrOpt) Path of the certfile for SSL.
enable = False	(BoolOpt) Toggle for SSL support on the keystone eventlet servers.
key_size = 1024	(IntOpt) SSL Key Length (in bits) (auto generated certificate).
keyfile = /etc/keystone/ssl/private/keystonekey.pem	(StrOpt) Path of the keyfile for SSL.
valid_days = 3650	(IntOpt) Days the certificate is valid for once signed (auto generated certificate).

Table 5.24. Description of configuration options for stats

Configuration option = Default value	Description
[stats]	
driver = keystone.contrib.stats.backends.kvs.Stats	(StrOpt) Keystone stats backend driver.

Table 5.25. Description of configuration options for token

Configuration option = Default value	Description
[token]	
bind =	(ListOpt) External auth mechanisms that should add bind information to token e.g. kerberos, x509.
cache_time = None	(IntOpt) Time to cache tokens (in seconds). This has no effect unless global and token caching are enabled.
caching = True	(BoolOpt) Toggle for token system caching. This has no effect unless global caching is enabled.
driver = keystone.token.backends.sql.Token	(StrOpt) Keystone Token persistence backend driver.
enforce_token_bind = permissive	(StrOpt) Enforcement policy on tokens presented to keystone with bind information. One of disabled, permissive, strict, required or a specifically required bind mode e.g. kerberos or x509 to require binding to that authentication.
expiration = 3600	(IntOpt) Amount of time a token should remain valid (in seconds).
provider = None	(StrOpt) Controls the token construction, validation, and revocation operations. Core providers are "keystone.token.providers.[pki uuid].Provider".
revocation_cache_time = 3600	(IntOpt) Time to cache the revocation list and the revocation events if revoke extension is enabled (in seconds). This has no effect unless global and token caching are enabled.
revoke_by_id = True	(BoolOpt) Revoke token by token identifier. Setting revoke_by_id to True enables various forms of enumerating tokens, e.g. `list tokens for user`. These enumerations are processed to determine the list of tokens to revoke. Only disable if you are switching to using the Revoke extension with a backend other than KVS, which stores events in memory.

Table 5.26. Description of configuration options for trust

Configuration option = Default value	Description
[trust]	
driver = keystone.trust.backends.sql.Trust	(StrOpt) Keystone Trust backend driver.
enabled = True	(BoolOpt) delegation and impersonation features can be optionally disabled.

Table 5.27. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
allowed_rpc_exception_modules = oslo.messaging.exceptions, nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.

Table 5.28. Description of configuration options for amqp

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
control_exchange = openstack	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
default_publisher_id = None	(StrOpt) Default publisher_id for outgoing notifications
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = notifications	(ListOpt) AMQP topic used for OpenStack notifications.
rpc_backend = rabbit	(StrOpt) The messaging driver to use, defaults to rabbit. Other drivers include qpid and zmq.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC greenthread pool.
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.

Table 5.29. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
qpid_password =	(StrOpt) Password for Qpid connection.
qpid_port = 5672	(IntOpt) Qpid broker port.
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for Qpid connection.

Table 5.30. Description of configuration options for rabbit

Configuration option = Default value	Description
[DEFAULT]	
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Configuration option = Default value	Description
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions.
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
rabbit_password = guest	(StrOpt) The RabbitMQ password.
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.

Table 5.31. Description of configuration options for zeromq

Configuration option = Default value	Description
[DEFAULT]	
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1.
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets.
rpc_zmq_matchmaker = oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver.
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port.
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Table 5.32. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	
host = 127.0.0.1	(StrOpt) Host to locate redis.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency.

Configuration option = Default value	Description
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).

Identity service sample configuration files

All the files in this section can be found in the `/etc/keystone` directory.

keystone.conf

The majority of the Identity service configuration is performed from the `keystone.conf` file.

```
[DEFAULT]

#
# Options defined in keystone
#

# A "shared secret" that can be used to bootstrap Keystone.
# This "token" does not represent a user, and carries no
# explicit authorization. To disable in production (highly
# recommended), remove AdminTokenAuthMiddleware from your
# paste application pipelines (for example, in keystone-
# paste.ini). (string value)
#admin_token=ADMIN

# The IP Address of the network interface to for the public
# service to listen on. (string value)
# Deprecated group/name - [DEFAULT]/bind_host
#public_bind_host=0.0.0.0

# The IP Address of the network interface to for the admin
# service to listen on. (string value)
# Deprecated group/name - [DEFAULT]/bind_host
#admin_bind_host=0.0.0.0

# The port which the OpenStack Compute service listens on.
# (integer value)
#compute_port=8774

# The port number which the admin service listens on. (integer
# value)
#admin_port=35357

# The port number which the public service listens on.
# (integer value)
#public_port=5000

# The base public endpoint URL for keystone that are
# advertised to clients (NOTE: this does NOT affect how
# keystone listens for connections) (string value).
# Defaults to the base host URL of the request. Eg a
```



```
# Sets the value of TCP_KEEPIDLE in seconds for each server
# socket. Only applies if tcp_keepalive is True. Not supported
# on OS X. (integer value)
#tcp_keepidle=600

# The maximum number of entities that will be returned in a
# collection can be set with list_limit, with no limit set by
# default. This global limit may be then overridden for a
# specific driver, by specifying a list_limit in the
# appropriate section (e.g. [assignment]). (integer value)
#list_limit=<None>

# Set this to false if you want to enable the ability for
# user, group and project entities to be moved between domains
# by updating their domain_id. Allowing such movement is not
# recommended if the scope of a domain admin is being
# restricted by use of an appropriate policy file (see
# policy.v3cloudsample as an example). (boolean value)
#domain_id_immutable=true

#
# Options defined in oslo.messaging
#

# Use durable queues in amqp. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_durable_queues
#amqp_durable_queues=false

# Auto-delete queues in amqp. (boolean value)
#amqp_auto_delete=false

# Size of RPC connection pool. (integer value)
#rpc_conn_pool_size=30

# Modules of exceptions that are permitted to be recreated
# upon receiving exception data from an rpc call. (list value)
#allowed_rpc_exception_modules=oslo.messaging.exceptions,nova.exception,
cinder.exception,exceptions

# Qpid broker hostname. (string value)
#qpid_hostname=localhost

# Qpid broker port. (integer value)
#qpid_port=5672

# Qpid HA cluster host:port pairs. (list value)
#qpid_hosts=$qpid_hostname:$qpid_port

# Username for Qpid connection. (string value)
#qpid_username=

# Password for Qpid connection. (string value)
#qpid_password=

# Space separated list of SASL mechanisms to use for auth.
# (string value)
#qpid_sasl_mechanisms=

# Seconds between connection keepalive heartbeats. (integer
```



```
#rabbit_virtual_host=/

# How frequently to retry connecting with RabbitMQ. (integer
# value)
#rabbit_retry_interval=1

# How long to backoff for between retries when connecting to
# RabbitMQ. (integer value)
#rabbit_retry_backoff=2

# Maximum number of RabbitMQ connection retries. Default is 0
# (infinite retry count). (integer value)
#rabbit_max_retries=0

# Use HA queues in RabbitMQ (x-ha-policy: all). If you change
# this option, you must wipe the RabbitMQ database. (boolean
# value)
#rabbit_ha_queues=false

# If passed, use a fake RabbitMQ provider. (boolean value)
#fake_rabbit=false

# ZeroMQ bind address. Should be a wildcard (*), an ethernet
# interface, or IP. The "host" option should point or resolve
# to this address. (string value)
#rpc_zmq_bind_address=*

# MatchMaker driver. (string value)
#rpc_zmq_matchmaker=oslo.messaging._drivers.matchmaker.MatchMakerLocalhost

# ZeroMQ receiver listening port. (integer value)
#rpc_zmq_port=9501

# Number of ZeroMQ contexts, defaults to 1. (integer value)
#rpc_zmq_contexts=1

# Maximum number of ingress messages to locally buffer per
# topic. Default is unlimited. (integer value)
#rpc_zmq_topic_backlog=<None>

# Directory for holding IPC sockets. (string value)
#rpc_zmq_ipc_dir=/var/run/openstack

# Name of this node. Must be a valid hostname, FQDN, or IP
# address. Must match "host" option, if running Nova. (string
# value)
#rpc_zmq_host=keystone

# Seconds to wait before a cast expires (TTL). Only supported
# by impl_zmq. (integer value)
#rpc_cast_timeout=30

# Heartbeat frequency. (integer value)
#matchmaker_heartbeat_freq=300

# Heartbeat time-to-live. (integer value)
#matchmaker_heartbeat_ttl=600

# Host to locate redis. (string value)
#host=127.0.0.1
```



```
# Data to append to log format when level is DEBUG (string
# value)
#logging_debug_format_suffix=%(funcName)s %(pathname)s:%(lineno)d

# Prefix each line of exception output with this format
# (string value)
#logging_exception_prefix=%(asctime)s.%(msecs)03d %(process)d TRACE %(name)s
%(instance)s

# List of logger=LEVEL pairs (list value)
#default_log_levels=amqp=WARN,amqpplib=WARN,boto=WARN,qpidd=WARN,sqlalchemy=
WARN,suds=INFO,iso8601=WARN,requests.packages.urllib3.connectionpool=WARN

# Publish error events (boolean value)
#publish_errors=false

# Make deprecations fatal (boolean value)
#fatal_deprecations=false

# If an instance is passed with the log message, format it
# like this (string value)
#instance_format="[instance: %(uuid)s] "

# If an instance UUID is passed with the log message, format
# it like this (string value)
#instance_uuid_format="[instance: %(uuid)s] "

# The name of logging configuration file. It does not disable
# existing loggers, but just appends specified logging
# configuration to any other existing logging options. Please
# see the Python logging module documentation for details on
# logging configuration files. (string value)
# Deprecated group/name - [DEFAULT]/log_config
#log_config_append=<None>

# DEPRECATED. A logging.Formatter log message format string
# which may use any of the available logging.LogRecord
# attributes. This option is deprecated. Please use
# logging_context_format_string and
# logging_default_format_string instead. (string value)
#log_format=<None>

# Format string for %(asctime)s in log records. Default:
# %(default)s (string value)
#log_date_format=%Y-%m-%d %H:%M:%S

# (Optional) Name of log file to output to. If no default is
# set, logging will go to stdout. (string value)
# Deprecated group/name - [DEFAULT]/logfile
#log_file=<None>

# (Optional) The base directory used for relative --log-file
# paths (string value)
# Deprecated group/name - [DEFAULT]/logdir
#log_dir=<None>

# Use syslog for logging. Existing syslog format is DEPRECATED
# during I, and then will be changed in J to honor RFC5424
# (boolean value)
```

```
#use_syslog=false

# (Optional) Use syslog rfc5424 format for logging. If
# enabled, will add APP-NAME (RFC5424) before the MSG part of
# the syslog message. The old format without APP-NAME is
# deprecated in I, and will be removed in J. (boolean value)
#use_syslog_rfc_format=false

# Syslog facility to receive log lines (string value)
#syslog_log_facility=LOG_USER

#
# Options defined in keystone.openstack.common.policy
#

# JSON file containing policy (string value)
#policy_file=policy.json

# Rule enforced when requested rule is not found (string
# value)
#policy_default_rule=default

[assignment]

#
# Options defined in keystone
#

# Keystone Assignment backend driver. (string value)
#driver=<None>

# Toggle for assignment caching. This has no effect unless
# global caching is enabled. (boolean value)
#caching=true

# TTL (in seconds) to cache assignment data. This has no
# effect unless global caching is enabled. (integer value)
#cache_time=<None>

# Maximum number of entities that will be returned in an
# assignment collection. (integer value)
#list_limit=<None>

[auth]

#
# Options defined in keystone
#

# Default auth methods. (list value)
#methods=external,password,token

# The password auth plugin module. (string value)
#password=keystone.auth.plugins.password.Password

# The token auth plugin module. (string value)
#token=keystone.auth.plugins.token.Token
```

```
# The external (REMOTE_USER) auth plugin module. (string
# value)
#external=keystone.auth.plugins.external.DefaultDomain

[cache]

#
# Options defined in keystone
#

# Prefix for building the configuration dictionary for the
# cache region. This should not need to be changed unless
# there is another dogpile.cache region with the same
# configuration name. (string value)
#config_prefix=cache.keystone

# Default TTL, in seconds, for any cached item in the
# dogpile.cache region. This applies to any cached method that
# doesn't have an explicit cache expiration time defined for
# it. (integer value)
#expiration_time=600

# Dogpile.cache backend module. It is recommended that
# Memcache (dogpile.cache.memcached) or Redis
# (dogpile.cache.redis) be used in production deployments.
# Small workloads (single process) like devstack can use the
# dogpile.cache.memory backend. (string value)
#backend=keystone.common.cache.noop

# Use a key-mangling function (shal) to ensure fixed length
# cache-keys. This is toggle-able for debugging purposes, it
# is highly recommended to always leave this set to True.
# (boolean value)
#use_key_mangler=true

# Arguments supplied to the backend module. Specify this
# option once per argument to be passed to the dogpile.cache
# backend. Example format: "<argname>:<value>". (multi valued)
#backend_argument=

# Proxy Classes to import that will affect the way the
# dogpile.cache backend functions. See the dogpile.cache
# documentation on changing-backend-behavior. Comma delimited
# list e.g. my.dogpile.proxy.Class, my.dogpile.proxyClass2.
# (list value)
#proxies=

# Global toggle for all caching using the should_cache_fn
# mechanism. (boolean value)
#enabled=false

# Extra debugging from the cache backend (cache keys,
# get/set/delete/etc calls) This is only really useful if you
# need to see the specific cache-backend get/set/delete calls
# with the keys/values. Typically this should be left set to
# False. (boolean value)
#debug_cache_backend=false
```

```
[catalog]

#
# Options defined in keystone
#

# Catalog template file name for use with the template catalog
# backend. (string value)
#template_file=default_catalog.templates

# Keystone catalog backend driver. (string value)
#driver=keystone.catalog.backends.sql.Catalog

# Maximum number of entities that will be returned in a
# catalog collection. (integer value)
#list_limit=<None>

[credential]

#
# Options defined in keystone
#

# Keystone Credential backend driver. (string value)
#driver=keystone.credential.backends.sql.Credential

[database]

#
# Options defined in keystone.openstack.common.db.options
#

# The file name to use with SQLite (string value)
#sqlite_db=keystone.sqlite

# If True, SQLite uses synchronous mode (boolean value)
#sqlite_synchronous=true

# The backend to use for db (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend=sqlalchemy

# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
#connection=<None>

# 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= (string
# value)
#mysql_sql_mode=TRADITIONAL

# Timeout before idle sql connections are reaped (integer
```



```
#db_inc_retry_interval=true

# max seconds between db connection retries, if
# db_inc_retry_interval is enabled (integer value)
#db_max_retry_interval=10

# maximum db connection retries before error is raised.
# (setting -1 implies an infinite retry count) (integer value)
#db_max_retries=20

[ec2]

#
# Options defined in keystone
#

# Keystone EC2Credential backend driver. (string value)
#driver=keystone.contrib.ec2.backends.kvs.Ec2

[endpoint_filter]

#
# Options defined in keystone
#

# Keystone Endpoint Filter backend driver (string value)
#driver=keystone.contrib.endpoint_filter.backends.sql.EndpointFilter

# Toggle to return all active endpoints if no filter exists.
# (boolean value)
#return_all_endpoints_if_no_filter=true

[federation]

#
# Options defined in keystone
#

# Keystone Federation backend driver. (string value)
#driver=keystone.contrib.federation.backends.sql.Federation

# Value to be used when filtering assertion parameters from
# the environment. (string value)
#assertion_prefix=

[identity]

#
# Options defined in keystone
#

# This references the domain to use for all Identity API v2
# requests (which are not aware of domains). A domain with
# this ID will be created for you by keystone-manage db_sync
# in migration 008. The domain referenced by this ID cannot
# be deleted on the v3 API, to prevent accidentally breaking
```

```
# the v2 API. There is nothing special about this domain,
# other than the fact that it must exist to order to maintain
# support for your v2 clients. (string value)
#default_domain_id=default

# A subset (or all) of domains can have their own identity
# driver, each with their own partial configuration file in a
# domain configuration directory. Only values specific to the
# domain need to be placed in the domain specific
# configuration file. This feature is disabled by default; set
# to True to enable. (boolean value)
#domain_specific_drivers_enabled=false

# Path for Keystone to locate the domain specific identity
# configuration files if domain_specific_drivers_enabled is
# set to true. (string value)
#domain_config_dir=/etc/keystone/domains

# Keystone Identity backend driver. (string value)
#driver=keystone.identity.backends.sql.Identity

# Maximum supported length for user passwords; decrease to
# improve performance. (integer value)
#max_password_length=4096

# Maximum number of entities that will be returned in an
# identity collection. (integer value)
#list_limit=<None>

[kvs]

#
# Options defined in keystone
#

# Extra dogpile.cache backend modules to register with the
# dogpile.cache library. (list value)
#backends=

# Prefix for building the configuration dictionary for the KVS
# region. This should not need to be changed unless there is
# another dogpile.cache region with the same configuration
# name. (string value)
#config_prefix=keystone.kvs

# Toggle to disable using a key-mangling function to ensure
# fixed length keys. This is toggle-able for debugging
# purposes, it is highly recommended to always leave this set
# to True. (boolean value)
#enable_key_mangler=true

# Default lock timeout for distributed locking. (integer
# value)
#default_lock_timeout=5

[ldap]

#
```

```
# Options defined in keystone
#

# URL for connecting to the LDAP server. (string value)
#url=ldap://localhost

# User BindDN to query the LDAP server. (string value)
#user=<None>

# Password for the BindDN to query the LDAP server. (string
# value)
#password=<None>

# LDAP server suffix (string value)
#suffix=cn=example,cn=com

# If true, will add a dummy member to groups. This is required
# if the objectclass for groups requires the "member"
# attribute. (boolean value)
#use_dumb_member=false

# DN of the "dummy member" to use when "use_dumb_member" is
# enabled. (string value)
#dumb_member=cn=dumb,dc=nonexistent

# allow deleting subtrees. (boolean value)
#allow_subtree_delete=false

# The LDAP scope for queries, this can be either "one"
# (onelevel/singleLevel) or "sub" (subtree/wholeSubtree).
# (string value)
#query_scope=one

# Maximum results per page; a value of zero ("0") disables
# paging. (integer value)
#page_size=0

# The LDAP dereferencing option for queries. This can be
# either "never", "searching", "always", "finding" or
# "default". The "default" option falls back to using default
# dereferencing configured by your ldap.conf. (string value)
#alias_dereferencing=default

# Override the system's default referral chasing behavior for
# queries. (boolean value)
#chase_referrals=<None>

# Search base for users. (string value)
#user_tree_dn=<None>

# LDAP search filter for users. (string value)
#user_filter=<None>

# LDAP objectClass for users. (string value)
#user_objectclass=inetOrgPerson

# LDAP attribute mapped to user id. (string value)
#user_id_attribute=cn

# LDAP attribute mapped to user name. (string value)
```

```
#user_name_attribute=sn

# LDAP attribute mapped to user email. (string value)
#user_mail_attribute=email

# LDAP attribute mapped to password. (string value)
#user_pass_attribute=userPassword

# LDAP attribute mapped to user enabled flag. (string value)
#user_enabled_attribute=enabled

# Bitmask integer to indicate the bit that the enabled value
# is stored in if the LDAP server represents "enabled" as a
# bit on an integer rather than a boolean. A value of "0"
# indicates the mask is not used. If this is not set to "0"
# the typical value is "2". This is typically used when
# "user_enabled_attribute = userAccountControl". (integer
# value)
#user_enabled_mask=0

# Default value to enable users. This should match an
# appropriate int value if the LDAP server uses non-boolean
# (bitmask) values to indicate if a user is enabled or
# disabled. If this is not set to "True" the typical value is
# "512". This is typically used when "user_enabled_attribute =
# userAccountControl". (string value)
#user_enabled_default=True

# List of attributes stripped off the user on update. (list
# value)
#user_attribute_ignore=default_project_id,tenants

# LDAP attribute mapped to default_project_id for users.
# (string value)
#user_default_project_id_attribute=<None>

# Allow user creation in LDAP backend. (boolean value)
#user_allow_create=true

# Allow user updates in LDAP backend. (boolean value)
#user_allow_update=true

# Allow user deletion in LDAP backend. (boolean value)
#user_allow_delete=true

# If True, Keystone uses an alternative method to determine if
# a user is enabled or not by checking if they are a member of
# the "user_enabled_emulation_dn" group. (boolean value)
#user_enabled_emulation=false

# DN of the group entry to hold enabled users when using
# enabled emulation. (string value)
#user_enabled_emulation_dn=<None>

# List of additional LDAP attributes used for mapping
# Additional attribute mappings for users. Attribute mapping
# format is <ldap_attr>:<user_attr>, where ldap_attr is the
# attribute in the LDAP entry and user_attr is the Identity
# API attribute. (list value)
#user_additional_attribute_mapping=
```

```
# Search base for projects (string value)
#tenant_tree_dn=<None>

# LDAP search filter for projects. (string value)
#tenant_filter=<None>

# LDAP objectClass for projects. (string value)
#tenant_objectclass=groupOfNames

# LDAP attribute mapped to project id. (string value)
#tenant_id_attribute=cn

# LDAP attribute mapped to project membership for user.
# (string value)
#tenant_member_attribute=member

# LDAP attribute mapped to project name. (string value)
#tenant_name_attribute=ou

# LDAP attribute mapped to project description. (string value)
#tenant_desc_attribute=description

# LDAP attribute mapped to project enabled. (string value)
#tenant_enabled_attribute=enabled

# LDAP attribute mapped to project domain_id. (string value)
#tenant_domain_id_attribute=businessCategory

# List of attributes stripped off the project on update. (list
# value)
#tenant_attribute_ignore=

# Allow tenant creation in LDAP backend. (boolean value)
#tenant_allow_create=true

# Allow tenant update in LDAP backend. (boolean value)
#tenant_allow_update=true

# Allow tenant deletion in LDAP backend. (boolean value)
#tenant_allow_delete=true

# If True, Keystone uses an alternative method to determine if
# a project is enabled or not by checking if they are a member
# of the "tenant_enabled_emulation_dn" group. (boolean value)
#tenant_enabled_emulation=false

# DN of the group entry to hold enabled projects when using
# enabled emulation. (string value)
#tenant_enabled_emulation_dn=<None>

# Additional attribute mappings for projects. Attribute
# mapping format is <ldap_attr>:<user_attr>, where ldap_attr
# is the attribute in the LDAP entry and user_attr is the
# Identity API attribute. (list value)
#tenant_additional_attribute_mapping=

# Search base for roles. (string value)
#role_tree_dn=<None>
```

```
# LDAP search filter for roles. (string value)
#role_filter=<None>

# LDAP objectClass for roles. (string value)
#role_objectclass=organizationalRole

# LDAP attribute mapped to role id. (string value)
#role_id_attribute=cn

# LDAP attribute mapped to role name. (string value)
#role_name_attribute=ou

# LDAP attribute mapped to role membership. (string value)
#role_member_attribute=roleOccupant

# List of attributes stripped off the role on update. (list
# value)
#role_attribute_ignore=

# Allow role creation in LDAP backend. (boolean value)
#role_allow_create=true

# Allow role update in LDAP backend. (boolean value)
#role_allow_update=true

# Allow role deletion in LDAP backend. (boolean value)
#role_allow_delete=true

# Additional attribute mappings for roles. Attribute mapping
# format is <ldap_attr>:<user_attr>, where ldap_attr is the
# attribute in the LDAP entry and user_attr is the Identity
# API attribute. (list value)
#role_additional_attribute_mapping=

# Search base for groups. (string value)
#group_tree_dn=<None>

# LDAP search filter for groups. (string value)
#group_filter=<None>

# LDAP objectClass for groups. (string value)
#group_objectclass=groupOfNames

# LDAP attribute mapped to group id. (string value)
#group_id_attribute=cn

# LDAP attribute mapped to group name. (string value)
#group_name_attribute=ou

# LDAP attribute mapped to show group membership. (string
# value)
#group_member_attribute=member

# LDAP attribute mapped to group description. (string value)
#group_desc_attribute=description

# List of attributes stripped off the group on update. (list
# value)
#group_attribute_ignore=
```

```
# Allow group creation in LDAP backend. (boolean value)
#group_allow_create=true

# Allow group update in LDAP backend. (boolean value)
#group_allow_update=true

# Allow group deletion in LDAP backend. (boolean value)
#group_allow_delete=true

# Additional attribute mappings for groups. Attribute mapping
# format is <ldap_attr>:<user_attr>, where ldap_attr is the
# attribute in the LDAP entry and user_attr is the Identity
# API attribute. (list value)
#group_additional_attribute_mapping=

# CA certificate file path for communicating with LDAP
# servers. (string value)
#tls_cacertfile=<None>

# CA certificate directory path for communicating with LDAP
# servers. (string value)
#tls_cacertdir=<None>

# Enable TLS for communicating with LDAP servers. (boolean
# value)
#use_tls=false

# valid options for tls_req_cert are demand, never, and allow.
# (string value)
#tls_req_cert=demand

[matchmaker_ring]

#
# Options defined in oslo.messaging
#

# Matchmaker ring file (JSON). (string value)
# Deprecated group/name - [DEFAULT]/matchmaker_ringfile
#ringfile=/etc/oslo/matchmaker_ring.json

[memcache]

#
# Options defined in keystone
#

# Memcache servers in the format of "host:port" (list value)
#servers=localhost:11211

# Number of compare-and-set attempts to make when using
# compare-and-set in the token memcache back end. (integer
# value)
#max_compare_and_set_retry=16

[oauth1]
```

```
#
# Options defined in keystone
#

# Keystone Credential backend driver. (string value)
#driver=keystone.contrib.oauth1.backends.sql.OAuth1

# Duration (in seconds) for the OAuth Request Token. (integer
# value)
#request_token_duration=28800

# Duration (in seconds) for the OAuth Access Token. (integer
# value)
#access_token_duration=86400

[os_inherit]

#
# Options defined in keystone
#

# role-assignment inheritance to projects from owning domain
# can be optionally enabled. (boolean value)
#enabled=false

[paste_deploy]

#
# Options defined in keystone
#

# Name of the paste configuration file that defines the
# available pipelines. (string value)
#config_file=keystone-paste.ini

[policy]

#
# Options defined in keystone
#

# Keystone Policy backend driver. (string value)
#driver=keystone.policy.backends.sql.Policy

# Maximum number of entities that will be returned in a policy
# collection. (integer value)
#list_limit=<None>

[revoke]

#
# Options defined in keystone
#

# An implementation of the backend for persisting revocation
# events. (string value)
```



```
#driver=keystone.contrib.revoke.backends.kvs.Revoke

# This value (calculated in seconds) is added to token
# expiration before a revocation event may be removed from the
# backend. (integer value)
#expiration_buffer=1800

# Toggle for revocation event cacheing. This has no effect
# unless global caching is enabled. (boolean value)
#caching=true

[signing]

#
# Options defined in keystone
#

# Deprecated in favor of provider in the [token] section.
# (string value)
#token_format=<None>

# Path of the certfile for token signing. (string value)
#certfile=/etc/keystone/ssl/certs/signing_cert.pem

# Path of the keyfile for token signing. (string value)
#keyfile=/etc/keystone/ssl/private/signing_key.pem

# Path of the CA for token signing. (string value)
#ca_certs=/etc/keystone/ssl/certs/ca.pem

# Path of the CA Key for token signing. (string value)
#ca_key=/etc/keystone/ssl/private/cakey.pem

# Key Size (in bits) for token signing cert (auto generated
# certificate). (integer value)
#key_size=2048

# Day the token signing cert is valid for (auto generated
# certificate). (integer value)
#valid_days=3650

# Certificate Subject (auto generated certificate) for token
# signing. (string value)
#cert_subject=/C=US/ST=Unset/L=Unset/O=Unset/CN=www.example.com

[ssl]

#
# Options defined in keystone
#

# Toggle for SSL support on the keystone eventlet servers.
# (boolean value)
#enable=false

# Path of the certfile for SSL. (string value)
#certfile=/etc/keystone/ssl/certs/keystone.pem
```



```
# Keystone Token persistence backend driver. (string value)
#driver=keystone.token.backends.sql.Token

# Toggle for token system cacheing. This has no effect unless
# global caching is enabled. (boolean value)
#caching=true

# Time to cache the revocation list and the revocation events
# if revoke extension is enabled (in seconds). This has no
# effect unless global and token caching are enabled. (integer
# value)
#revocation_cache_time=3600

# Time to cache tokens (in seconds). This has no effect unless
# global and token caching are enabled. (integer value)
#cache_time=<None>

# Revoke token by token identifier. Setting revoke_by_id to
# True enables various forms of enumerating tokens, e.g. `list
# tokens for user`. These enumerations are processed to
# determine the list of tokens to revoke. Only disable if
# you are switching to using the Revoke extension with a
# backend other than KVS, which stores events in memory.
# (boolean value)
#revoke_by_id=true

[trust]

#
# Options defined in keystone
#

# delegation and impersonation features can be optionally
# disabled. (boolean value)
#enabled=true

# Keystone Trust backend driver. (string value)
#driver=keystone.trust.backends.sql.Trust
```

keystone-paste.ini

The `keystone-paste.ini` file configures the Web Service Gateway Interface (WSGI) middleware pipeline for the Identity service.

```
# Keystone PasteDeploy configuration file.

[filter:debug]
paste.filter_factory = keystone.common.wsgi:Debug.factory

[filter:build_auth_context]
paste.filter_factory = keystone.middleware:AuthContextMiddleware.factory

[filter:token_auth]
```

```
paste.filter_factory = keystone.middleware:TokenAuthMiddleware.factory

[filter:admin_token_auth]
paste.filter_factory = keystone.middleware:AdminTokenAuthMiddleware.factory

[filter:xml_body]
paste.filter_factory = keystone.middleware:XmlBodyMiddleware.factory

[filter:xml_body_v2]
paste.filter_factory = keystone.middleware:XmlBodyMiddlewareV2.factory

[filter:xml_body_v3]
paste.filter_factory = keystone.middleware:XmlBodyMiddlewareV3.factory

[filter:json_body]
paste.filter_factory = keystone.middleware:JsonBodyMiddleware.factory

[filter:user_crud_extension]
paste.filter_factory = keystone.contrib.user_crud:CrudExtension.factory

[filter:crud_extension]
paste.filter_factory = keystone.contrib.admin_crud:CrudExtension.factory

[filter:ec2_extension]
paste.filter_factory = keystone.contrib.ec2:Ec2Extension.factory

[filter:ec2_extension_v3]
paste.filter_factory = keystone.contrib.ec2:Ec2ExtensionV3.factory

[filter:federation_extension]
paste.filter_factory = keystone.contrib.federation.
routers:FederationExtension.factory

[filter:oauth1_extension]
paste.filter_factory = keystone.contrib.oauth1.routers:OAuth1Extension.factory

[filter:s3_extension]
paste.filter_factory = keystone.contrib.s3:S3Extension.factory

[filter:endpoint_filter_extension]
paste.filter_factory = keystone.contrib.endpoint_filter.
routers:EndpointFilterExtension.factory

[filter:simple_cert_extension]
paste.filter_factory = keystone.contrib.simple_cert:SimpleCertExtension.
factory

[filter:revoke_extension]
paste.filter_factory = keystone.contrib.revoke.routers:RevokeExtension.factory

[filter:url_normalize]
paste.filter_factory = keystone.middleware:NormalizingFilter.factory

[filter:sizelimit]
paste.filter_factory = keystone.middleware:RequestBodySizeLimiter.factory

[filter:stats_monitoring]
paste.filter_factory = keystone.contrib.stats:StatsMiddleware.factory

[filter:stats_reporting]
```


logging.conf

A special logging configuration file can be specified in the `keystone.conf` configuration file (for example, `/etc/keystone/logging.conf`). For details, see the Python logging module documentation ([Python Logging](#)).

```
[loggers]
keys=root,access

[handlers]
keys=production,file,access_file,devel

[formatters]
keys=minimal,normal,debug

#####
# Loggers #
#####

[logger_root]
level=WARNING
handlers=file

[logger_access]
level=INFO
qualname=access
handlers=access_file

#####
# Log Handlers #
#####

[handler_production]
class=handlers.SysLogHandler
level=ERROR
formatter=normal
args=( 'localhost', handlers.SYSLOG_UDP_PORT), handlers.SysLogHandler.
LOG_USER)

[handler_file]
class=handlers.WatchedFileHandler
level=WARNING
formatter=normal
args=( 'error.log', )

[handler_access_file]
class=handlers.WatchedFileHandler
level=INFO
formatter=minimal
args=( 'access.log', )

[handler_devel]
class=StreamHandler
level=NOTSET
formatter=debug
args=( sys.stdout, )
```


Option = default value	(Type) Help string
domain_id_immutable = True	(BoolOpt) Set this to false if you want to enable the ability for user, group and project entities to be moved between domains by updating their domain_id. Allowing such movement is not recommended if the scope of a domain admin is being restricted by use of an appropriate policy file (see policy.v3cloudsample as an example).
host = 127.0.0.1	(StrOpt) Host to locate redis.
keystone_ec2_cafile = None	(StrOpt) A PEM encoded certificate authority to use when verifying HTTPS connections. Defaults to the system CAs.
keystone_ec2_certfile = None	(StrOpt) Client certificate key filename. Required if EC2 server requires client certificate.
keystone_ec2_insecure = False	(BoolOpt) Disable SSL certificate verification.
keystone_ec2_keyfile = None	(StrOpt) Required if EC2 server requires client certificate.
keystone_ec2_url = http://localhost:5000/v2.0/ec2tokens	(StrOpt) URL to get token from ec2 request.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
list_limit = None	(IntOpt) The maximum number of entities that will be returned in a collection can be set with list_limit, with no limit set by default. This global limit may be then overridden for a specific driver, by specifying a list_limit in the appropriate section (e.g. [assignment]).
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.
public_bind_host = 0.0.0.0	(StrOpt) The IP Address of the network interface to for the public service to listen on.
rabbit_login_method = AMQPPLAIN	(StrOpt) the RabbitMQ login method
tcp_keepalive = False	(BoolOpt) Set this to True if you want to enable TCP_KEEPAIVE on server sockets i.e. sockets used by the keystone wsgi server for client connections.
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIIDLE in seconds for each server socket. Only applies if tcp_keepalive is True. Not supported on OS X.
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
assignment/list_limit = None	(IntOpt) Maximum number of entities that will be returned in an assignment collection.
audit/namespace = openstack	(StrOpt) namespace prefix for generated id
catalog/list_limit = None	(IntOpt) Maximum number of entities that will be returned in a catalog collection.
database/db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
database/db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)

Option	Previous default value	New default value
		suds=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s %(user_identity)s] %(instance)s%(message)s
public_endpoint	http://localhost:%(public_port)s/	None
rpc_zmq_matchmaker	keystone.openstack.common.rpc.matchmaker	keystone.matchmaker.matchmaker.MatchMakerLocalhost
auth/external	keystone.auth.plugins.external.ExternalDomain	keystone.auth.plugins.external.DefaultDomain
database/connection	sqlite:///home/gauvain/sources/ openstack/openstack-doc-tools/ autogenerate_config_docs/sources/ keystone/keystone/openstack/ common/db/\$sqlite_db	None
ldap/group_additional_attribute_mapping	None	
ldap/group_attribute_ignore		
ldap/role_additional_attribute_mapping	None	
ldap/role_attribute_ignore		
ldap/tenant_additional_attribute_mapping	None	
ldap/tenant_attribute_ignore		
ldap/user_additional_attribute_mapping	None	
ldap/user_attribute_ignore	default_project_id,tenants	default_project_id, tenants
memcache/servers	localhost:11211	localhost:11211
paste_deploy/config_file	None	keystone-paste.ini
signing/ca_key	/etc/keystone/ssl/certs/cakey.pem	/etc/keystone/ssl/private/cakey.pem
ssl/ca_key	/etc/keystone/ssl/certs/cakey.pem	/etc/keystone/ssl/private/cakey.pem
token/expiration	86400	3600

Table 5.35. Deprecated options

Deprecated option	New Option
bind_host	admin_bind_host
log_config	log_config_append
rpc_notifier2/topics	notification_topics
sql/connection	database/connection
sql/idle_timeout	database/idle_timeout
bind_host	public_bind_host

6. Image Service

Table of Contents

Configure the API	336
Configure the RPC messaging system	338
Support for ISO images	340
Configuring Backends	341
Image Service sample configuration files	346
New, updated and deprecated options for glance	365

Compute relies on an external image service to store virtual machine images and maintain a catalog of available images. By default, Compute is configured to use the OpenStack Image Service (Glance), which is currently the only supported image service.

If your installation requires euca2ools to register new images, you must run the `nova-objectstore` service. This service provides an Amazon S3 front-end for Glance, which is required by euca2ools.

To customize the Compute Service, use the configuration option settings documented in [Table 2.27, "Description of configuration options for glance" \[214\]](#) and [Table 2.47, "Description of configuration options for s3" \[227\]](#).

You can modify many options in the OpenStack Image Service. The following tables provide a comprehensive list.

Table 6.1. Description of configuration options for auth_token

Configuration option = Default value	Description
[DEFAULT]	
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPS connections. Defaults to system CAs.

Configuration option = Default value	Description
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) 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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. 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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

Table 6.2. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
allow_additional_image_properties = True	(BoolOpt) Whether to allow users to specify image properties beyond what the image schema provides
api_limit_max = 1000	(IntOpt) Maximum permissible number of items that could be returned by a request

Configuration option = Default value	Description
backlog = 4096	(IntOpt) The backlog value that will be used when creating the TCP listener socket.
bind_host = 0.0.0.0	(StrOpt) Address to bind the server. Useful when selecting a particular network interface.
bind_port = None	(IntOpt) The port on which the server will listen.
data_api = glance.db.sqlalchemy.api	(StrOpt) Python module path of data access API
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
image_location_quota = 10	(IntOpt) Maximum number of locations allowed on an image. Negative values evaluate to unlimited.
image_member_quota = 128	(IntOpt) Maximum number of image members per image. Negative values evaluate to unlimited.
image_property_quota = 128	(IntOpt) Maximum number of properties allowed on an image. Negative values evaluate to unlimited.
image_tag_quota = 128	(IntOpt) Maximum number of tags allowed on an image. Negative values evaluate to unlimited.
limit_param_default = 25	(IntOpt) Default value for the number of items returned by a request if not specified explicitly in the request
lock_path = None	(StrOpt) Directory to use for lock files.
metadata_encryption_key = None	(StrOpt) Key used for encrypting sensitive metadata while talking to the registry or database.
notifier_strategy = default	(StrOpt) Notifications can be sent when images are create, updated or deleted. There are three methods of sending notifications, logging (via the log_file directive), rabbit (via a rabbitmq queue), qpid (via a Qpid message queue), or noop (no notifications sent, the default). (DEPRECATED)
os_region_name = None	(StrOpt) Region name of this node.
property_protection_file = None	(StrOpt) The location of the property protection file.
property_protection_rule_format = roles	(StrOpt) This config value indicates whether "roles" or "policies" are used in the property protection file.
show_image_direct_url = False	(BoolOpt) Whether to include the backend image storage location in image properties. Revealing storage location can be a security risk, so use this setting with caution!
user_storage_quota = 0	(IntOpt) Set a system wide quota for every user. This value is the total number of bytes that a user can use across all storage systems. A value of 0 means unlimited.
workers = 1	(IntOpt) The number of child process workers that will be created to service API requests.
[image_format]	
container_formats = ami, ari, aki, bare, ovf, ova	(ListOpt) Supported values for the 'container_format' image attribute
disk_formats = ami, ari, aki, vhd, vmdk, raw, qcow2, vdi, iso	(ListOpt) Supported values for the 'disk_format' image attribute
[task]	
task_time_to_live = 48	(IntOpt) Time in hours for which a task lives after, either succeeding or failing

Table 6.3. Description of configuration options for db

Configuration option = Default value	Description
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db

Configuration option = Default value	Description
connection = None	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
db_retry_interval = 1	(IntOpt) seconds between db connection retries
idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
mysql_sql_mode = TRADITIONAL	(StrOpt) 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=
pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
sqlite_db = glance.sqlite	(StrOpt) The file name to use with SQLite
sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost

Table 6.4. Description of configuration options for imagecache

Configuration option = Default value	Description
[DEFAULT]	
cleanup_scrubber = False	(BoolOpt) A boolean that determines if the scrubber should clean up the files it uses for taking data. Only one server in your deployment should be designated the cleanup host.
cleanup_scrubber_time = 86400	(IntOpt) Items must have a modified time that is older than this value in order to be candidates for cleanup.
delayed_delete = False	(BoolOpt) Turn on/off delayed delete.
image_cache_dir = None	(StrOpt) Base directory that the Image Cache uses.
image_cache_driver = sqlite	(StrOpt) The driver to use for image cache management.
image_cache_max_size = 10737418240	(IntOpt) The maximum size in bytes that the cache can use.
image_cache_sqlite_db = cache.db	(StrOpt) The path to the sqlite file database that will be used for image cache management.

Configuration option = Default value	Description
image_cache_stall_time = 86400	(IntOpt) The amount of time to let an image remain in the cache without being accessed.
scrub_time = 0	(IntOpt) The amount of time in seconds to delay before performing a delete.
scrubber_datadir = /var/lib/glance/scrubber	(StrOpt) Directory that the scrubber will use to track information about what to delete. Make sure this is set in glance-api.conf and glance-scrubber.conf.

Table 6.5. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
default_log_levels = amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN	(ListOpt) List of logger=LEVEL pairs
fatal_deprecations = False	(BoolOpt) Make deprecations fatal
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s. %(msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s %(user_identity)s] %(instance)s %(message)s	(StrOpt) Format string to use for log messages with context
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) Data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) Format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) Prefix each line of exception output with this format
publish_errors = False	(BoolOpt) Publish error events
syslog_log_facility = LOG_USER	(StrOpt) Syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error

Configuration option = Default value	Description
use_syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and then will be changed in J to honor RFC5424
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 6.6. Description of configuration options for matchmaker

Configuration option = Default value	Description
[DEFAULT]	
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON).

Table 6.7. Description of configuration options for paste

Configuration option = Default value	Description
[paste_deploy]	
config_file = None	(StrOpt) Name of the paste configuration file.
flavor = None	(StrOpt) Partial name of a pipeline in your paste configuration file with the service name removed. For example, if your paste section name is [pipeline:glance-api-keystone] use the value "keystone"

Table 6.8. Description of configuration options for policy

Configuration option = Default value	Description
[DEFAULT]	
policy_default_rule = default	(StrOpt) The default policy to use.
policy_file = policy.json	(StrOpt) The location of the policy file.

Table 6.9. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	
host = 127.0.0.1	(StrOpt) Host to locate redis.
password = None	(StrOpt) Password for Redis server (optional).
port = 6379	(IntOpt) Use this port to connect to redis host.

Table 6.10. Description of configuration options for registry

Configuration option = Default value	Description
[DEFAULT]	
admin_password = None	(StrOpt) The administrators password.
admin_tenant_name = None	(StrOpt) The tenant name of the administrative user.
admin_user = None	(StrOpt) The administrators user name.

Configuration option = Default value	Description
auth_region = None	(StrOpt) The region for the authentication service.
auth_strategy = noauth	(StrOpt) The strategy to use for authentication.
auth_url = None	(StrOpt) The URL to the keystone service.
registry_client_ca_file = None	(StrOpt) The path to the certifying authority cert file to use in SSL connections to the registry server.
registry_client_cert_file = None	(StrOpt) The path to the cert file to use in SSL connections to the registry server.
registry_client_insecure = False	(BoolOpt) When using SSL in connections to the registry server, do not require validation via a certifying authority.
registry_client_key_file = None	(StrOpt) The path to the key file to use in SSL connections to the registry server.
registry_client_protocol = http	(StrOpt) The protocol to use for communication with the registry server. Either http or https.
registry_client_timeout = 600	(IntOpt) The period of time, in seconds, that the API server will wait for a registry request to complete. A value of 0 implies no timeout.
registry_host = 0.0.0.0	(StrOpt) Address to find the registry server.
registry_port = 9191	(IntOpt) Port the registry server is listening on.

Table 6.11. Description of configuration options for testing

Configuration option = Default value	Description
[DEFAULT]	
pydev_worker_debug_host = None	(StrOpt) The hostname/IP of the pydev process listening for debug connections
pydev_worker_debug_port = 5678	(IntOpt) The port on which a pydev process is listening for connections.

Configure the API

The Image Service has two APIs: the user-facing API, and the registry API, which is for internal requests that require access to the database.

Both of the APIs currently have two major versions, v1 and v2. It is possible to run either or both version, by setting appropriate values of `enable_v1_api`, `enable_v2_api`, `enable_v1_registry` and `enable_v2_registry`. If the v2 API is used, running `glance-registry` is optional, as v2 of `glance-api` can connect directly to the database.

Tables of all options used to configure the APIs, including enabling SSL and modifying WSGI settings are found below.

Table 6.12. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
admin_role = admin	(StrOpt) Role used to identify an authenticated user as administrator.
allow_anonymous_access = False	(BoolOpt) Allow unauthenticated users to access the API with read-only privileges. This only applies when using ContextMiddleware.
default_publisher_id = image.localhost	(StrOpt) Default publisher_id for outgoing notifications.

Configuration option = Default value	Description
default_store = file	(StrOpt) Default scheme to use to store image data. The scheme must be registered by one of the stores defined by the 'known_stores' config option.
enable_v1_api = True	(BoolOpt) Deploy the v1 OpenStack Images API.
enable_v1_registry = True	(BoolOpt) Deploy the v1 OpenStack Registry API.
enable_v2_api = True	(BoolOpt) Deploy the v2 OpenStack Images API.
enable_v2_registry = True	(BoolOpt) Deploy the v2 OpenStack Registry API.
image_size_cap = 109951162776	(IntOpt) Maximum size of image a user can upload in bytes. Defaults to 109951162776 bytes (1 TB).
known_stores = glance.store.filesystem.Store, glance.store.http.Store	(ListOpt) List of which store classes and store class locations are currently known to glance at startup.
location_strategy = location_order	(StrOpt) This value sets what strategy will be used to determine the image location order. Currently two strategies are packaged with Glance 'location_order' and 'store_type'.
owner_is_tenant = True	(BoolOpt) When true, this option sets the owner of an image to be the tenant. Otherwise, the owner of the image will be the authenticated user issuing the request.
send_identity_headers = False	(BoolOpt) Whether to pass through headers containing user and tenant information when making requests to the registry. This allows the registry to use the context middleware without the keystoneclients' auth_token middleware, removing calls to the keystone auth service. It is recommended that when using this option, secure communication between glance api and glance registry is ensured by means other than auth_token middleware.
show_multiple_locations = False	(BoolOpt) Whether to include the backend image locations in image properties. Revealing storage location can be a security risk, so use this setting with caution! The overrides show_image_direct_url.
use_user_token = True	(BoolOpt) Whether to pass through the user token when making requests to the registry.
[store_type_location_strategy]	
store_type_preference =	(ListOpt) The store names to use to get store preference order. The name must be registered by one of the stores defined by the 'known_stores' config option. This option will be applied when you using 'store_type' option as image location strategy defined by the 'location_strategy' config option.

Table 6.13. Description of configuration options for ssl

Configuration option = Default value	Description
[DEFAULT]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients.
cert_file = None	(StrOpt) Certificate file to use when starting API server securely.
key_file = None	(StrOpt) Private key file to use when starting API server securely.

Table 6.14. Description of configuration options for wsgi

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
backdoor_port = None	(IntOpt) port for eventlet backdoor to listen
eventlet_hub = poll	(StrOpt) Name of eventlet hub to use. Traditionally, we have only supported 'poll', however 'selects' may be appropriate for some platforms. See http://eventlet.net/doc/hubs.html for more details.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs)
tcp_keepidle = 600	(IntOpt) The value for the socket option TCP_KEEPIDLE. This is the time in seconds that the connection must be idle before TCP starts sending keepalive probes.

Configure the RPC messaging system

OpenStack projects use an open standard for messaging middleware known as AMQP. This messaging middleware enables the OpenStack services that run on multiple servers to talk to each other. The OpenStack common library project, oslo, supports three implementations of AMQP: RabbitMQ, Qpid, and ZeroMQ.

The following tables contain settings to configure the messaging middleware for the Image Service:

Table 6.15. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
control_exchange = openstack	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider.
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions.
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
rabbit_login_method = AMQPLAIN	(StrOpt) the RabbitMQ login method
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
rabbit_password = guest	(StrOpt) The RabbitMQ password.
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.

Configuration option = Default value	Description
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.

Table 6.16. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
qpid_password =	(StrOpt) Password for Qpid connection.
qpid_port = 5672	(IntOpt) Qpid broker port.
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for Qpid connection.

Table 6.17. Description of configuration options for zmq

Configuration option = Default value	Description
[DEFAULT]	
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1.
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets.
rpc_zmq_matchmaker = oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver.
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port.
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Table 6.18. Description of configuration options for amqp

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = notifications	(ListOpt) AMQP topic used for OpenStack notifications.
rpc_backend = rabbit	(StrOpt) The messaging driver to use, defaults to rabbit. Other drivers include qpid and zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC greenthread pool.
transport_url = None	(StrOpt) A URL representing the messaging driver to use and its full configuration. If not set, we fall back to the rpc_backend option and driver specific configuration.

Table 6.19. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
allowed_rpc_exception_modules = openstack.common.exception, glance.common.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.

Support for ISO images

You can load ISO images into the Image Service. You can subsequently boot an ISO image using Compute.

Procedure 6.1. To load an ISO image to an Image Service data store

1. Obtain the ISO image. For example, `ubuntu-13.04-server-amd64.iso`.
2. In the Image Service, run the following command:

```
$ glance image-create --name ubuntu.iso \  
  --is-public=True --container-format=bare \  
  --disk-format=iso < ubuntu-13.04-server-amd64.iso
```

In this command, `ubuntu.iso` is the name for the ISO image after it is loaded to the Image Service, and `ubuntu-13.04-server-amd64.iso` is the name of the source ISO image.

3. Optionally, confirm the upload in Compute.

Run this command:

```
$ nova image-list
```

Procedure 6.2. To boot an instance from an ISO image

- Run this command:

```
$ nova boot --image ubuntu.iso \
  --flavor 1 instance_name
```

In this command, `ubuntu.iso` is the ISO image, and `instance_name` is the name of the new instance.

Configuring Backends

The image service supports several different backends for storing virtual machine images, including Cinder, a directory on a local file system, GridFS, Ceph RBD, Amazon S3, Sheepdog, OpenStack Object Storage or VMWare ESX. The following tables detail the options available for each.

Table 6.20. Description of configuration options for cinder

Configuration option = Default value	Description
[DEFAULT]	
<code>cinder_api_insecure = False</code>	(BoolOpt) Allow to perform insecure SSL requests to cinder.
<code>cinder_ca_certificates_file = None</code>	(StrOpt) Location of CA certificates file to use for cinder client requests.
<code>cinder_catalog_info = volume:cinder:publicURL</code>	(StrOpt) Info to match when looking for cinder in the service catalog. Format is: separated values of the form: <code><service_type>:<service_name>:<endpoint_type></code> .
<code>cinder_endpoint_template = None</code>	(StrOpt) Override service catalog lookup with template for cinder endpoint e.g. <code>http://localhost:8776/v1/%(project_id)s</code> .
<code>cinder_http_retries = 3</code>	(IntOpt) Number of cinderclient retries on failed http calls.

Table 6.21. Description of configuration options for filesystem

Configuration option = Default value	Description
[DEFAULT]	
<code>filesystem_store_datadir = None</code>	(StrOpt) Directory to which the Filesystem backend store writes images.
<code>filesystem_store_datadirs = None</code>	(MultiStrOpt) List of directories and its priorities to which the Filesystem backend store writes images.
<code>filesystem_store_metadata_file = None</code>	(StrOpt) The path to a file which contains the metadata to be returned with any location associated with this store. The file must contain a valid JSON dict.

Table 6.22. Description of configuration options for gridfs

Configuration option = Default value	Description
[DEFAULT]	
<code>mongodb_store_db = None</code>	(StrOpt) Database to use.
<code>mongodb_store_uri = None</code>	(StrOpt) Hostname or IP address of the instance to connect to, or a mongodb URI, or a list of hostnames / mongodb URIs. If host is an IPv6 literal it must be enclosed in '[' and ']' characters following the RFC2732 URL syntax (e.g. '[::1]' for localhost).

Table 6.23. Description of configuration options for rbd

Configuration option = Default value	Description
[DEFAULT]	
rbdd_store_ceph_conf = /etc/ceph/ceph.conf	(StrOpt) Ceph configuration file path. If <None>, librados will locate the default config. If using cephx authentication, this file should include a reference to the right keyring in a client.<USER> section.
rbdd_store_chunk_size = 8	(IntOpt) RADOS images will be chunked into objects of this size (in megabytes). For best performance, this should be a power of two.
rbdd_store_pool = images	(StrOpt) RADOS pool in which images are stored.
rbdd_store_user = None	(StrOpt) RADOS user to authenticate as (only applicable if using Cephx. If <None>, a default will be chosen based on the client. section in rbd_store_ceph_conf).

Table 6.24. Description of configuration options for s3

Configuration option = Default value	Description
[DEFAULT]	
s3_store_access_key = None	(StrOpt) The S3 query token access key.
s3_store_bucket = None	(StrOpt) The S3 bucket to be used to store the Glance data.
s3_store_bucket_url_format = subdomain	(StrOpt) The S3 calling format used to determine the bucket. Either subdomain or path can be used.
s3_store_create_bucket_on_put = False	(BoolOpt) A boolean to determine if the S3 bucket should be created on upload if it does not exist or if an error should be returned to the user.
s3_store_host = None	(StrOpt) The host where the S3 server is listening.
s3_store_object_buffer_dir = None	(StrOpt) The local directory where uploads will be staged before they are transferred into S3.
s3_store_secret_key = None	(StrOpt) The S3 query token secret key.

Table 6.25. Description of configuration options for sheepdog

Configuration option = Default value	Description
[DEFAULT]	
sheepdog_store_address = 127.0.0.1	(StrOpt) IP address of sheep daemon.
sheepdog_store_chunk_size = 64	(IntOpt) Images will be chunked into objects of this size (in megabytes). For best performance, this should be a power of two.
sheepdog_store_port = 7000	(IntOpt) Port of sheep daemon.

Table 6.26. Description of configuration options for swift

Configuration option = Default value	Description
[DEFAULT]	
swift_enable_snet = False	(BoolOpt) Whether to use ServiceNET to communicate with the Swift storage servers.
swift_store_admin_tenants =	(ListOpt) A list of tenants that will be granted read/write access on all Swift containers created by Glance in multi-tenant mode.
swift_store_auth_address = None	(StrOpt) The address where the Swift authentication service is listening.

Configuration option = Default value	Description
<code>swift_store_auth_insecure = False</code>	(BoolOpt) If True, swiftclient won't check for a valid SSL certificate when authenticating.
<code>swift_store_auth_version = 2</code>	(StrOpt) Version of the authentication service to use. Valid versions are 2 for keystone and 1 for swauth and rackspace.
<code>swift_store_container = glance</code>	(StrOpt) Container within the account that the account should use for storing images in Swift.
<code>swift_store_create_container_on_put = False</code>	(BoolOpt) A boolean value that determines if we create the container if it does not exist.
<code>swift_store_endpoint_type = publicURL</code>	(StrOpt) A string giving the endpoint type of the swift service to use (publicURL, adminURL or internalURL). This setting is only used if <code>swift_store_auth_version</code> is 2.
<code>swift_store_key = None</code>	(StrOpt) Auth key for the user authenticating against the Swift authentication service.
<code>swift_store_large_object_chunk_size = 200</code>	(IntOpt) The amount of data written to a temporary disk buffer during the process of chunking the image file.
<code>swift_store_large_object_size = 5120</code>	(IntOpt) The size, in MB, that Glance will start chunking image files and do a large object manifest in Swift.
<code>swift_store_multi_tenant = False</code>	(BoolOpt) If set to True, enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts.
<code>swift_store_region = None</code>	(StrOpt) The region of the swift endpoint to be used for single tenant. This setting is only necessary if the tenant has multiple swift endpoints.
<code>swift_store_retry_get_count = 0</code>	(IntOpt) The number of times a Swift download will be retried before the request fails.
<code>swift_store_service_type = object-store</code>	(StrOpt) A string giving the service type of the swift service to use. This setting is only used if <code>swift_store_auth_version</code> is 2.
<code>swift_store_ssl_compression = True</code>	(BoolOpt) If set to False, disables SSL layer compression of https swift requests. Setting to False may improve performance for images which are already in a compressed format, eg qcow2.
<code>swift_store_user = None</code>	(StrOpt) The user to authenticate against the Swift authentication service.

Configure vCenter data stores for the Image Service back end

To use vCenter data stores for the Image Service back end, you must update the `glance-api.conf` file, as follows:

- Add data store parameters to the `VMware Datastore Store Options` section.
- Specify vSphere as the back end.



Note

You must configure any configured Image Service data stores for the Compute service.

You can specify vCenter data stores directly by using the data store name or Storage Policy Based Management (SPBM), which requires vCenter Server 5.5 or later. For details, see [the section called “Configure vCenter data stores for the back end” \[345\]](#).

**Note**

If you intend to use multiple data stores for the back end, use the SPBM feature.

In the `DEFAULT` section, set the `default_store` parameter to `vsphere`, as shown in this code sample:

```
[DEFAULT]
# Which back end scheme should Glance use by default is not specified
# in a request to add a new image to Glance? Known schemes are determined
# by the known_stores option below.
# Default: 'file'
default_store = vsphere
```

The following table describes the parameters in the `VMware Datastore Store Options` section:

Table 6.27. Description of configuration options for vmware

Configuration option = Default value	Description
[DEFAULT]	
vmware_api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to ESX/VC.
vmware_api_retry_count = 10	(IntOpt) Number of times VMware ESX/VC server API must be retried upon connection related issues.
vmware_datacenter_path = ha-datacenter	(StrOpt) Inventory path to a datacenter. If the <code>vmware_server_host</code> specified is an ESX/ESXi, the <code>vmware_datacenter_path</code> is optional. If specified, it should be "ha-datacenter".
vmware_datastore_name = None	(StrOpt) Datastore associated with the datacenter.
vmware_server_host = None	(StrOpt) ESX/ESXi or vCenter Server target system. The server value can be an IP address or a DNS name.
vmware_server_password = None	(StrOpt) Password for authenticating with VMware ESX/VC server.
vmware_server_username = None	(StrOpt) Username for authenticating with VMware ESX/VC server.
vmware_store_image_dir = /openstack_glance	(StrOpt) The name of the directory where the glance images will be stored in the VMware datastore.
vmware_task_poll_interval = 5	(IntOpt) The interval used for polling remote tasks invoked on VMware ESX/VC server.

The following block of text shows a sample configuration:

```
# ===== VMware Datastore Store Options =====
# ESX/ESXi or vCenter Server target system.
# The server value can be an IP address or a DNS name
# e.g. 127.0.0.1, 127.0.0.1:443, www.vmware-infra.com
vmware_server_host = 192.168.0.10

# Server username (string value)
vmware_server_username = ADMINISTRATOR

# Server password (string value)
vmware_server_password = password

# Inventory path to a datacenter (string value)
```

```
# Value optional when vmware_server_ip is an ESX/ESXi host: if specified
# should be `ha-datacenter`.
vmware_datacenter_path = DATACENTER

# Datastore associated with the datacenter (string value)
vmware_datastore_name = datastore1

# PBM service WSDL file location URL. e.g.
# file:///opt/SDK/spbm/wsd/pbmService.wsdl Not setting this
# will disable storage policy based placement of images.
# (string value)
vmware_pbm_wsd_location =

# The PBM policy. If `pbm_wsd_location` is set, a PBM policy needs
# to be specified. This policy will be used to select the datastore
# in which the images will be stored.
vmware_pbm_policy =

# The interval used for polling remote tasks
# invoked on VMware ESX/VC server in seconds (integer value)
vmware_task_poll_interval = 5

# Absolute path of the folder containing the images in the datastore
# (string value)
vmware_store_image_dir = /openstack_glance

# Allow to perform insecure SSL requests to the target system (boolean value)
vmware_api_insecure = False
```

Configure vCenter data stores for the back end

You can specify a vCenter data store for the back end by setting the `vmware_datastore_name` parameter value to the vCenter name of the data store. This configuration limits the back end to a single data store.

Alternatively, you can specify a SPBM policy, which can comprise multiple vCenter data stores. Both approaches are described.



Note

SPBM requires vCenter Server 5.5 or later.

Procedure 6.3. To configure a single data store

1. If present, comment or delete the `vmware_pbm_wsd_location` and `vmware_pbm_policy` parameters.
2. Uncomment and define the `vmware_datastore_name` parameter with the name of the vCenter data store.
3. Complete the other vCenter configuration parameters as appropriate.

Procedure 6.4. To configure multiple data stores using SPBM

1. In vCenter, use tagging to identify the data stores and define a storage policy:
 - a. Create the tag.

- b. Apply the tag to the data stores to be used by the SPBM policy.
- c. Create a tag-based storage policy that uses one or more tags to identify a set of data stores.

**Note**

For details about creating tags in vSphere, see the [vSphere documentation](#).

For details about storage policies in vSphere, see the [vSphere documentation](#).

2. Return to the `glance-api.conf` file.
3. Comment or delete the `vmware_datastore_name` parameter.
4. Uncomment and define the `vmware_pbm_policy` parameter by entering the same value as the tag you defined and applied to the data stores in vCenter.
5. Uncomment and define the `vmware_pbm_wsdll_location` parameter by entering the location of the PBM service WSDL file. For example, `file:///opt/SDK/spbm/wsdll/pbmService.wsdl`.

**Note**

If you do not set this parameter, the storage policy cannot be used to place images in the data store.

Complete the other vCenter configuration parameters as appropriate.

Image Service sample configuration files

All the files in this section can be found in the `/etc/glance/` directory.

glance-api.conf

The configuration file for the Image Service API is found in the `glance-api.conf` file.

This file must be modified after installation.

```
[DEFAULT]
# Show more verbose log output (sets INFO log level output)
#verbose = False

# Show debugging output in logs (sets DEBUG log level output)
#debug = False

# Which backend scheme should Glance use by default is not specified
# in a request to add a new image to Glance? Known schemes are determined
# by the known_stores option below.
# Default: 'file'
default_store = file

# List of which store classes and store class locations are
# currently known to glance at startup.
```

```
# Existing but disabled stores:
#   glance.store.rbd.Store,
#   glance.store.s3.Store,
#   glance.store.swift.Store,
#   glance.store.sheepdog.Store,
#   glance.store.cinder.Store,
#   glance.store.gridfs.Store,
#   glance.store.vmware_datastore.Store,
#known_stores = glance.store.filesystem.Store,
#               glance.store.http.Store

# Maximum image size (in bytes) that may be uploaded through the
# Glance API server. Defaults to 1 TB.
# WARNING: this value should only be increased after careful consideration
# and must be set to a value under 8 EB (9223372036854775808).
#image_size_cap = 1099511627776

# Address to bind the API server
bind_host = 0.0.0.0

# Port the bind the API server to
bind_port = 9292

# Log to this file. Make sure you do not set the same log file for both the
# API
# and registry servers!
#
# If `log_file` is omitted and `use_syslog` is false, then log messages are
# sent to stdout as a fallback.
log_file = /var/log/glance/api.log

# Backlog requests when creating socket
backlog = 4096

# TCP_KEEPIIDLE value in seconds when creating socket.
# Not supported on OS X.
#tcp_keepidle = 600

# API to use for accessing data. Default value points to sqlalchemy
# package, it is also possible to use: glance.db.registry.api
# data_api = glance.db.sqlalchemy.api

# Number of Glance API worker processes to start.
# On machines with more than one CPU increasing this value
# may improve performance (especially if using SSL with
# compression turned on). It is typically recommended to set
# this value to the number of CPUs present on your machine.
workers = 1

# Maximum line size of message headers to be accepted.
# max_header_line may need to be increased when using large tokens
# (typically those generated by the Keystone v3 API with big service
# catalogs)
# max_header_line = 16384

# Role used to identify an authenticated user as administrator
#admin_role = admin

# Allow unauthenticated users to access the API with read-only
```

```
# privileges. This only applies when using ContextMiddleware.
#allow_anonymous_access = False

# Allow access to version 1 of glance api
#enable_v1_api = True

# Allow access to version 2 of glance api
#enable_v2_api = True

# Return the URL that references where the data is stored on
# the backend storage system. For example, if using the
# file system store a URL of 'file:///path/to/image' will
# be returned to the user in the 'direct_url' meta-data field.
# The default value is false.
#show_image_direct_url = False

# Send headers containing user and tenant information when making requests to
# the v1 glance registry. This allows the registry to function as if a user is
# authenticated without the need to authenticate a user itself using the
# auth_token middleware.
# The default value is false.
#send_identity_headers = False

# Supported values for the 'container_format' image attribute
#container_formats=ami,ari,aki,bare,ovf,ova

# Supported values for the 'disk_format' image attribute
#disk_formats=ami,ari,aki,vhd,vmdk,raw,qcow2,vdi,iso

# Directory to use for lock files. Default to a temp directory
# (string value). This setting needs to be the same for both
# glance-scrubber and glance-api.
#lock_path=<None>

# Property Protections config file
# This file contains the rules for property protections and the roles/policies
# associated with it.
# If this config value is not specified, by default, property protections
# won't be enforced.
# If a value is specified and the file is not found, then the glance-api
# service will not start.
#property_protection_file =

# Specify whether 'roles' or 'policies' are used in the
# property_protection_file.
# The default value for property_protection_rule_format is 'roles'.
#property_protection_rule_format = roles

# Specifies how long (in hours) a task is supposed to live in the tasks DB
# after succeeding or failing before getting soft-deleted.
# The default value for task_time_to_live is 48 hours.
# task_time_to_live = 48

# This value sets what strategy will be used to determine the image location
# order. Currently two strategies are packaged with Glance 'location_order'
# and 'store_type'.
#location_strategy = location_order

# ===== Syslog Options =====
```

```
# Send logs to syslog (/dev/log) instead of to file specified
# by `log_file`
#use_syslog = False

# Facility to use. If unset defaults to LOG_USER.
#syslog_log_facility = LOG_LOCAL0

# ===== SSL Options =====

# Certificate file to use when starting API server securely
#cert_file = /path/to/certfile

# Private key file to use when starting API server securely
#key_file = /path/to/keyfile

# CA certificate file to use to verify connecting clients
#ca_file = /path/to/cafile

# ===== Security Options =====

# AES key for encrypting store 'location' metadata, including
# -- if used -- Swift or S3 credentials
# Should be set to a random string of length 16, 24 or 32 bytes
#metadata_encryption_key = <16, 24 or 32 char registry metadata key>

# ===== Registry Options =====

# Address to find the registry server
registry_host = 0.0.0.0

# Port the registry server is listening on
registry_port = 9191

# What protocol to use when connecting to the registry server?
# Set to https for secure HTTP communication
registry_client_protocol = http

# The path to the key file to use in SSL connections to the
# registry server, if any. Alternately, you may set the
# GLANCE_CLIENT_KEY_FILE environ variable to a filepath of the key file
#registry_client_key_file = /path/to/key/file

# The path to the cert file to use in SSL connections to the
# registry server, if any. Alternately, you may set the
# GLANCE_CLIENT_CERT_FILE environ variable to a filepath of the cert file
#registry_client_cert_file = /path/to/cert/file

# The path to the certifying authority cert file to use in SSL connections
# to the registry server, if any. Alternately, you may set the
# GLANCE_CLIENT_CA_FILE environ variable to a filepath of the CA cert file
#registry_client_ca_file = /path/to/ca/file

# When using SSL in connections to the registry server, do not require
# validation via a certifying authority. This is the registry's equivalent of
# specifying --insecure on the command line using glanceclient for the API
# Default: False
#registry_client_insecure = False

# The period of time, in seconds, that the API server will wait for a registry
# request to complete. A value of '0' implies no timeout.
```



```
# Configuration options if sending notifications via Qpid (these are
# the defaults)
qpid_notification_exchange = glance
qpid_notification_topic = notifications
qpid_hostname = localhost
qpid_port = 5672
qpid_username =
qpid_password =
qpid_sasl_mechanisms =
qpid_reconnect_timeout = 0
qpid_reconnect_limit = 0
qpid_reconnect_interval_min = 0
qpid_reconnect_interval_max = 0
qpid_reconnect_interval = 0
qpid_heartbeat = 5
# Set to 'ssl' to enable SSL
qpid_protocol = tcp
qpid_tcp_nodelay = True

# ===== Filesystem Store Options =====

# Directory that the Filesystem backend store
# writes image data to
filesystem_store_datadir = /var/lib/glance/images/

# A list of directories where image data can be stored.
# This option may be specified multiple times for specifying multiple store
# directories. Either one of filesystem_store_datadirs or
# filesystem_store_datadir option is required. A priority number may be given
# after each directory entry, separated by a ":".
# When adding an image, the highest priority directory will be selected,
# unless
# there is not enough space available in cases where the image size is already
# known. If no priority is given, it is assumed to be zero and the directory
# will be considered for selection last. If multiple directories have the same
# priority, then the one with the most free space available is selected.
# If same store is specified multiple times then BadStoreConfiguration
# exception will be raised.
#filesystem_store_datadirs = /var/lib/glance/images/:1

# A path to a JSON file that contains metadata describing the storage
# system. When show_multiple_locations is True the information in this
# file will be returned with any location that is contained in this
# store.
#filesystem_store_metadata_file = None

# ===== Swift Store Options =====

# Version of the authentication service to use
# Valid versions are '2' for keystone and '1' for swauth and raxspace
swift_store_auth_version = 2

# Address where the Swift authentication service lives
# Valid schemes are 'http://' and 'https://'
# If no scheme specified, default to 'https://'
# For swauth, use something like '127.0.0.1:8080/v1.0/'
swift_store_auth_address = 127.0.0.1:5000/v2.0/

# User to authenticate against the Swift authentication service
# If you use Swift authentication service, set it to 'account':'user'
```

```
# where 'account' is a Swift storage account and 'user'
# is a user in that account
swift_store_user = jdoe:jdoe

# Auth key for the user authenticating against the
# Swift authentication service
swift_store_key = a86850deb2742ec3cb41518e26aa2d89

# Container within the account that the account should use
# for storing images in Swift
swift_store_container = glance

# Do we create the container if it does not exist?
swift_store_create_container_on_put = False

# What size, in MB, should Glance start chunking image files
# and do a large object manifest in Swift? By default, this is
# the maximum object size in Swift, which is 5GB
swift_store_large_object_size = 5120

# When doing a large object manifest, what size, in MB, should
# Glance write chunks to Swift? This amount of data is written
# to a temporary disk buffer during the process of chunking
# the image file, and the default is 200MB
swift_store_large_object_chunk_size = 200

# Whether to use ServiceNET to communicate with the Swift storage servers.
# (If you aren't RACKSPACE, leave this False!)
#
# To use ServiceNET for authentication, prefix hostname of
# `swift_store_auth_address` with 'snet-'.
# Ex. https://example.com/v1.0/ -> https://snet-example.com/v1.0/
swift_enable_snet = False

# If set to True enables multi-tenant storage mode which causes Glance images
# to be stored in tenant specific Swift accounts.
#swift_store_multi_tenant = False

# A list of swift ACL strings that will be applied as both read and
# write ACLs to the containers created by Glance in multi-tenant
# mode. This grants the specified tenants/users read and write access
# to all newly created image objects. The standard swift ACL string
# formats are allowed, including:
# <tenant_id>:<username>
# <tenant_name>:<username>
# *:<username>
# Multiple ACLs can be combined using a comma separated list, for
# example: swift_store_admin_tenants = service:glance,*:admin
#swift_store_admin_tenants =

# The region of the swift endpoint to be used for single tenant. This setting
# is only necessary if the tenant has multiple swift endpoints.
#swift_store_region =

# If set to False, disables SSL layer compression of https swift requests.
# Setting to 'False' may improve performance for images which are already
# in a compressed format, eg qcow2. If set to True, enables SSL layer
# compression (provided it is supported by the target swift proxy).
#swift_store_ssl_compression = True
```



```

# invoked on VMware ESX/VC server in seconds (integer value)
#vmware_task_poll_interval = 5

# Absolute path of the folder containing the images in the datastore
# (string value)
#vmware_store_image_dir = /openstack_glance

# Allow to perform insecure SSL requests to the target system (boolean value)
#vmware_api_insecure = False

# ===== Delayed Delete Options =====

# Turn on/off delayed delete
delayed_delete = False

# Delayed delete time in seconds
scrub_time = 43200

# Directory that the scrubber will use to remind itself of what to delete
# Make sure this is also set in glance-scrubber.conf
scrubber_data_dir = /var/lib/glance/scrubber

# ===== Quota Options =====

# The maximum number of image members allowed per image
#image_member_quota = 128

# The maximum number of image properties allowed per image
#image_property_quota = 128

# The maximum number of tags allowed per image
#image_tag_quota = 128

# The maximum number of locations allowed per image
#image_location_quota = 10

# Set a system wide quota for every user. This value is the total number
# of bytes that a user can use across all storage systems. A value of
# 0 means unlimited.
#user_storage_quota = 0

# ===== Image Cache Options =====

# Base directory that the Image Cache uses
image_cache_dir = /var/lib/glance/image-cache/

# ===== Manager Options =====

# DEPRECATED. TO BE REMOVED IN THE JUNO RELEASE.
# Whether or not to enforce that all DB tables have charset utf8.
# If your database tables do not have charset utf8 you will
# need to convert before this option is removed. This option is
# only relevant if your database engine is MySQL.
#db_enforce_mysql_charset = True

# ===== Database Options =====

[database]
# The file name to use with SQLite (string value)
#sqlite_db = glance.sqlite

```

```
# If True, SQLite uses synchronous mode (boolean value)
#sqlite_synchronous = True

# The backend to use for db (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
#connection = <None>

# 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= (string
# value)
#mysql_sql_mode = TRADITIONAL

# Timeout before idle sql connections are reaped (integer
# value)
# Deprecated group/name - [DEFAULT]/sql_idle_timeout
# Deprecated group/name - [DATABASE]/sql_idle_timeout
# Deprecated group/name - [sql]/idle_timeout
#idle_timeout = 3600

# Minimum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_min_pool_size
# Deprecated group/name - [DATABASE]/sql_min_pool_size
#min_pool_size = 1

# Maximum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_pool_size
# Deprecated group/name - [DATABASE]/sql_max_pool_size
#max_pool_size = <None>

# Maximum db connection retries during startup. (setting -1
# implies an infinite retry count) (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_retries
# Deprecated group/name - [DATABASE]/sql_max_retries
#max_retries = 10

# Interval between retries of opening a sql connection
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_retry_interval
# Deprecated group/name - [DATABASE]/reconnect_interval
#retry_interval = 10

# If set, use this value for max_overflow with sqlalchemy
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_overflow
# Deprecated group/name - [DATABASE]/sqlalchemy_max_overflow
#max_overflow = <None>

# Verbosity of SQL debugging information. 0=None,
```


glance-registry.conf

Configuration for the Image Service's registry, which stores the metadata about images, is found in the `glance-registry.conf` file.

This file must be modified after installation.

```
[DEFAULT]
# Show more verbose log output (sets INFO log level output)
#verbose = False

# Show debugging output in logs (sets DEBUG log level output)
#debug = False

# Address to bind the registry server
bind_host = 0.0.0.0

# Port the bind the registry server to
bind_port = 9191

# Log to this file. Make sure you do not set the same log file for both the
  API
# and registry servers!
#
# If `log_file` is omitted and `use_syslog` is false, then log messages are
# sent to stdout as a fallback.
log_file = /var/log/glance/registry.log

# Backlog requests when creating socket
backlog = 4096

# TCP_KEEPIDLE value in seconds when creating socket.
# Not supported on OS X.
#tcp_keepidle = 600

# API to use for accessing data. Default value points to sqlalchemy
# package.
#data_api = glance.db.sqlalchemy.api

# Enable Registry API versions individually or simultaneously
#enable_v1_registry = True
#enable_v2_registry = True

# Limit the api to return `param_limit_max` items in a call to a container. If
# a larger `limit` query param is provided, it will be reduced to this value.
api_limit_max = 1000

# If a `limit` query param is not provided in an api request, it will
# default to `limit_param_default`
limit_param_default = 25

# Role used to identify an authenticated user as administrator
#admin_role = admin

# Whether to automatically create the database tables.
# Default: False
#db_auto_create = False

# Enable DEBUG log messages from sqlalchemy which prints every database
```



```
# Deprecated group/name - [DEFAULT]/sql_min_pool_size
# Deprecated group/name - [DATABASE]/sql_min_pool_size
#min_pool_size = 1

# Maximum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_pool_size
# Deprecated group/name - [DATABASE]/sql_max_pool_size
#max_pool_size = <None>

# Maximum db connection retries during startup. (setting -1
# implies an infinite retry count) (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_retries
# Deprecated group/name - [DATABASE]/sql_max_retries
#max_retries = 10

# Interval between retries of opening a sql connection
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_retry_interval
# Deprecated group/name - [DATABASE]/reconnect_interval
#retry_interval = 10

# If set, use this value for max_overflow with sqlalchemy
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_overflow
# Deprecated group/name - [DATABASE]/sqlalchemy_max_overflow
#max_overflow = <None>

# Verbosity of SQL debugging information. 0=None,
# 100=Everything (integer value)
# Deprecated group/name - [DEFAULT]/sql_connection_debug
#connection_debug = 0

# Add python stack traces to SQL as comment strings (boolean
# value)
# Deprecated group/name - [DEFAULT]/sql_connection_trace
#connection_trace = False

# If set, use this value for pool_timeout with sqlalchemy
# (integer value)
# Deprecated group/name - [DATABASE]/sqlalchemy_pool_timeout
#pool_timeout = <None>

# Enable the experimental use of database reconnect on
# connection lost (boolean value)
#use_db_reconnect = False

# seconds between db connection retries (integer value)
#db_retry_interval = 1

# Whether to increase interval between db connection retries,
# up to db_max_retry_interval (boolean value)
#db_inc_retry_interval = True

# max seconds between db connection retries, if
# db_inc_retry_interval is enabled (integer value)
#db_max_retry_interval = 10

# maximum db connection retries before error is raised.
# (setting -1 implies an infinite retry count) (integer value)
```

```
#db_max_retries = 20

[keystone_auth token]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%

[paste_deploy]
# Name of the paste configuration file that defines the available pipelines
#config_file = glance-registry-paste.ini

# Partial name of a pipeline in your paste configuration file with the
# service name removed. For example, if your paste section name is
# [pipeline:glance-registry-keystone], you would configure the flavor below
# as 'keystone'.
#flavor=
```

glance-api-paste.ini

Configuration for the Image Service's API middleware pipeline is found in the `glance-api-paste.ini` file.

You should not need to modify this file.

```
# Use this pipeline for no auth or image caching - DEFAULT
[pipeline:glance-api]
pipeline = versionnegotiation unauthenticated-context rootapp

# Use this pipeline for image caching and no auth
[pipeline:glance-api-caching]
pipeline = versionnegotiation unauthenticated-context cache rootapp

# Use this pipeline for caching w/ management interface but no auth
[pipeline:glance-api-cachemanagement]
pipeline = versionnegotiation unauthenticated-context cache cachemanage
rootapp

# Use this pipeline for keystone auth
[pipeline:glance-api-keystone]
pipeline = versionnegotiation auth token context rootapp

# Use this pipeline for keystone auth with image caching
[pipeline:glance-api-keystone+caching]
pipeline = versionnegotiation auth token context cache rootapp

# Use this pipeline for keystone auth with caching and cache management
[pipeline:glance-api-keystone+cachemanagement]
pipeline = versionnegotiation auth token context cache cachemanage rootapp

# Use this pipeline for authZ only. This means that the registry will treat a
# user as authenticated without making requests to keystone to reauthenticate
# the user.
[pipeline:glance-api-trusted-auth]
pipeline = versionnegotiation context rootapp

# Use this pipeline for authZ only. This means that the registry will treat a
# user as authenticated without making requests to keystone to reauthenticate
```

```
# the user and uses cache management
[pipeline:glance-api-trusted-auth+cachemanagement]
pipeline = versionnegotiation context cache cachemanage rootapp

[composite:rootapp]
paste.composite_factory = glance.api:root_app_factory
/: apiversions
/v1: apiv1app
/v2: apiv2app

[app:apiversions]
paste.app_factory = glance.api.versions:create_resource

[app:apiv1app]
paste.app_factory = glance.api.v1.router:API.factory

[app:apiv2app]
paste.app_factory = glance.api.v2.router:API.factory

[filter:versionnegotiation]
paste.filter_factory = glance.api.middleware.
version_negotiation:VersionNegotiationFilter.factory

[filter:cache]
paste.filter_factory = glance.api.middleware.cache:CacheFilter.factory

[filter:cachemanage]
paste.filter_factory = glance.api.middleware.cache_manage:CacheManageFilter.
factory

[filter:context]
paste.filter_factory = glance.api.middleware.context:ContextMiddleware.factory

[filter:unauthenticated-context]
paste.filter_factory = glance.api.middleware.
context:UnauthenticatedContextMiddleware.factory

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
delay_auth_decision = true

[filter:gzip]
paste.filter_factory = glance.api.middleware.gzip:GzipMiddleware.factory
```

glance-registry-paste.ini

The Image Service's middleware pipeline for its registry is found in the `glance-registry-paste.ini` file.

```
# Use this pipeline for no auth - DEFAULT
[pipeline:glance-registry]
pipeline = unauthenticated-context registryapp

# Use this pipeline for keystone auth
[pipeline:glance-registry-keystone]
pipeline = authtoken context registryapp

# Use this pipeline for authZ only. This means that the registry will treat a
# user as authenticated without making requests to keystone to reauthenticate
```

```
# the user.
[pipeline:glance-registry-trusted-auth]
pipeline = context registryapp

[app:registryapp]
paste.app_factory = glance.registry.api:API.factory

[filter:context]
paste.filter_factory = glance.api.middleware.context:ContextMiddleware.factory

[filter:unauthenticated-context]
paste.filter_factory = glance.api.middleware.
context:UnauthenticatedContextMiddleware.factory

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
```

glance-scrubber.conf

glance-scrubber is a utility for the Image Service that cleans up images that have been deleted; its configuration is stored in the `glance-scrubber.conf` file.

Multiple instances of `glance-scrubber` can be run in a single deployment, but only one of them can be designated as the `cleanup_scrubber` in the `glance-scrubber.conf` file. The `cleanup_scrubber` coordinates other `glance-scrubber` instances by maintaining the master queue of images that need to be removed.

```
[DEFAULT]
# Show more verbose log output (sets INFO log level output)
#verbose = False

# Show debugging output in logs (sets DEBUG log level output)
#debug = False

# Log to this file. Make sure you do not set the same log file for both the
# API
# and registry servers!
#
# If `log_file` is omitted and `use_syslog` is false, then log messages are
# sent to stdout as a fallback.
log_file = /var/log/glance/scrubber.log

# Send logs to syslog (/dev/log) instead of to file specified by `log_file`
#use_syslog = False

# Should we run our own loop or rely on cron/scheduler to run us
daemon = False

# Loop time between checking for new items to schedule for delete
wakeup_time = 300

# Directory that the scrubber will use to remind itself of what to delete
# Make sure this is also set in glance-api.conf
scrubber_datadir = /var/lib/glance/scrubber

# Only one server in your deployment should be designated the cleanup host
cleanup_scrubber = False

# pending_delete items older than this time are candidates for cleanup
```



```

    "get_tasks": "",
    "add_task": "",
    "modify_task": ""
}

```

New, updated and deprecated options for glance

Table 6.28. New options

Option = default value	(Type) Help string
enable_v1_registry = True	(BoolOpt) Deploy the v1 OpenStack Registry API.
enable_v2_registry = True	(BoolOpt) Deploy the v2 OpenStack Registry API.
filesystem_store_datadirs = None	(MultiStrOpt) List of directories and its priorities to which the Filesystem backend store writes images.
image_location_quota = 10	(IntOpt) Maximum number of locations allowed on an image. Negative values evaluate to unlimited.
image_member_quota = 128	(IntOpt) Maximum number of image members per image. Negative values evaluate to unlimited.
image_property_quota = 128	(IntOpt) Maximum number of properties allowed on an image. Negative values evaluate to unlimited.
image_tag_quota = 128	(IntOpt) Maximum number of tags allowed on an image. Negative values evaluate to unlimited.
location_strategy = location_order	(StrOpt) This value sets what strategy will be used to determine the image location order. Currently two strategies are packaged with Glance 'location_order' and 'store_type'.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs
property_protection_rule_format = roles	(StrOpt) This config value indicates whether "roles" or "policies" are used in the property protection file.
swift_store_retry_get_count = 0	(IntOpt) The number of times a Swift download will be retried before the request fails.
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
vmware_api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to ESX/VC.
vmware_api_retry_count = 10	(IntOpt) Number of times VMware ESX/VC server API must be retried upon connection related issues.
vmware_datacenter_path = ha-datacenter	(StrOpt) Inventory path to a datacenter. If the vmware_server_host specified is an ESX/ESXi, the vmware_datacenter_path is optional. If specified, it should be "ha-datacenter".
vmware_datastore_name = None	(StrOpt) Datastore associated with the datacenter.
vmware_server_host = None	(StrOpt) ESX/ESXi or vCenter Server target system. The server value can be an IP address or a DNS name.

Option = default value	(Type) Help string
vmware_server_password = None	(StrOpt) Password for authenticating with VMware ESX/VC server.
vmware_server_username = None	(StrOpt) Username for authenticating with VMware ESX/VC server.
vmware_store_image_dir = /openstack_glance	(StrOpt) The name of the directory where the glance images will be stored in the VMware datastore.
vmware_task_poll_interval = 5	(IntOpt) The interval used for polling remote tasks invoked on VMware ESX/VC server.
database/backend = sqlalchemy	(StrOpt) The backend to use for db
database/connection = None	(StrOpt) The SQLAlchemy connection string used to connect to the database
database/connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
database/connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
database/db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
database/db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
database/db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
database/db_retry_interval = 1	(IntOpt) seconds between db connection retries
database/idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
database/max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
database/max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
database/max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
database/min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
database/mysql_sql_mode = TRADITIONAL	(StrOpt) 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=
database/pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
database/retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
database/sqlite_db = glance.sqlite	(StrOpt) The file name to use with SQLite
database/sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
database/use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost
image_format/container_formats = ['ami', 'ari', 'aki', 'bare', 'ovf', 'ova']	(ListOpt) Supported values for the 'container_format' image attribute
image_format/disk_formats = ['ami', 'ari', 'aki', 'vhd', 'vmdk', 'raw', 'qcow2', 'vdi', 'iso']	(ListOpt) Supported values for the 'disk_format' image attribute
store_type_location_strategy/store_type_preference = []	(ListOpt) The store names to use to get store preference order. The name must be registered by one of the stores defined by the 'known_stores' config option. This option will be applied when you using 'store_type' option as

Option = default value	(Type) Help string image location strategy defined by the 'location_strategy' config option.
task/task_time_to_live = 48	(IntOpt) Time in hours for which a task lives after, either succeeding or failing

Table 6.29. New default values

Option	Previous default value	New default value
default_log_levels	amqpplib=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, eventlet.wsgi.server=WARN	amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN
known_stores	glance.store.filesystem.Store, glance.store.http.Store, glance.store.rbd.Store, glance.store.s3.Store, glance.store.swift.Store, glance.store.sheepdog.Store, glance.store.cinder.Store	glance.store.filesystem.Store, glance.store.http.Store
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%request_id)s %(user_identity)s] %(instance)s%(message)s
qpid_port	5672	5672
rbd_store_ceph_conf		/etc/ceph/ceph.conf
rbd_store_chunk_size	4	8
rbd_store_pool	rbd	images
sheepdog_store_port	7000	7000

Table 6.30. Deprecated options

Deprecated option	New Option
rabbit_durable_queues	amqp_durable_queues
disk_formats	image_format/disk_formats
log_config	log_config_append
sql_connection	database/connection
sql_max_retries	database/max_retries
sql_retry_interval	database/retry_interval
sql_idle_timeout	database/idle_timeout
container_formats	image_format/container_formats

7. Networking

Table of Contents

Networking configuration options	368
Log files used by Networking	400
Networking sample configuration files	401
New, updated and deprecated options for neutron	418

This chapter explains the OpenStack Networking configuration options. For installation prerequisites, steps, and use cases, see the *OpenStack Installation Guide* for your distribution (docs.openstack.org) and [Cloud Administrator Guide](#).

Networking configuration options

The options and descriptions listed in this introduction are auto generated from the code in the Networking service project, which provides software-defined networking between VMs run in Compute. The list contains common options, while the subsections list the options for the various networking plug-ins.

Table 7.1. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
admin_password = None	(StrOpt) Admin password
admin_tenant_name = None	(StrOpt) Admin tenant name
admin_user = None	(StrOpt) Admin username
agent_down_time = 75	(IntOpt) Seconds to regard the agent is down; should be at least twice report_interval, to be sure the agent is down for good.
allowed_rpc_exception_modules = nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.
api_workers = 0	(IntOpt) Number of separate worker processes for service
auth_ca_cert = None	(StrOpt) Certificate Authority public key (CA cert) file for ssl
auth_insecure = False	(BoolOpt) Turn off verification of the certificate for ssl
auth_region = None	(StrOpt) Authentication region
auth_strategy = keystone	(StrOpt) The type of authentication to use
auth_url = None	(StrOpt) Authentication URL
base_mac = fa:16:3e:00:00:00	(StrOpt) The base MAC address Neutron will use for VIFs
bind_host = 0.0.0.0	(StrOpt) The host IP to bind to
bind_port = 9696	(IntOpt) The port to bind to
ca_certs = None	(StrOpt) CA certificates
core_plugin = None	(StrOpt) The core plugin Neutron will use
ctl_cert = None	(StrOpt) controller certificate
ctl_privkey = None	(StrOpt) controller private key

Configuration option = Default value	Description
dhcp_agent_notification = True	(BoolOpt) Allow sending resource operation notification to DHCP agent
dhcp_agents_per_network = 1	(IntOpt) Number of DHCP agents scheduled to host a network.
dhcp_confs = \$state_path/dhcp	(StrOpt) Location to store DHCP server config files
dhcp_delete_namespaces = False	(BoolOpt) Delete namespace after removing a dhcp server.
dhcp_domain = openstacklocal	(StrOpt) Domain to use for building the hostnames
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq	(StrOpt) The driver used to manage the DHCP server.
dhcp_lease_duration = 86400	(IntOpt) DHCP lease duration
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
endpoint_type = adminURL	(StrOpt) Network service endpoint type to pull from the keystone catalog
force_gateway_on_subnet = False	(BoolOpt) Ensure that configured gateway is on subnet
interface_driver = None	(StrOpt) The driver used to manage the virtual interface.
ip_lib_force_root = False	(BoolOpt) Force ip_lib calls to use the root helper
lock_path = None	(StrOpt) Directory to use for lock files.
mac_generation_retries = 16	(IntOpt) How many times Neutron will retry MAC generation
max_dns_nameservers = 5	(IntOpt) Maximum number of DNS nameservers
max_fixed_ips_per_port = 5	(IntOpt) Maximum number of fixed ips per port
max_subnet_host_routes = 20	(IntOpt) Maximum number of host routes per subnet
periodic_fuzzy_delay = 5	(IntOpt) Range of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = 40	(IntOpt) Seconds between running periodic tasks
report_interval = 300	(IntOpt) Interval between two metering reports
root_helper = sudo	(StrOpt) Root helper application.
state_path = /var/lib/neutron	(StrOpt) Where to store Neutron state files. This directory must be writable by the agent.
[AGENT]	
root_helper = sudo	(StrOpt) Root helper application.
[PROXY]	
admin_password = None	(StrOpt) Admin password
admin_tenant_name = None	(StrOpt) Admin tenant name
admin_user = None	(StrOpt) Admin user
auth_region = None	(StrOpt) Authentication region
auth_strategy = keystone	(StrOpt) The type of authentication to use
auth_url = None	(StrOpt) Authentication URL
[heleos]	
admin_password = None	(StrOpt) ESM admin password.

Networking plug-ins

OpenStack Networking introduces the concept of a plug-in, which is a back-end implementation of the OpenStack Networking API. A plug-in can use a variety of technologies to implement the logical API requests. Some OpenStack Networking plug-ins might use basic Linux VLANs and IP tables, while others might use more

advanced technologies, such as L2-in-L3 tunneling or OpenFlow. These sections detail the configuration options for the various plug-ins.

BigSwitch configuration options

Table 7.2. Description of configuration options for bigswitch

Configuration option = Default value	Description
[NOVA]	
node_override_vif_802.1qbg =	(ListOpt) Nova compute nodes to manually set VIF type to 802.1qbg
node_override_vif_802.1qbh =	(ListOpt) Nova compute nodes to manually set VIF type to 802.1qbh
node_override_vif_binding_failed =	(ListOpt) Nova compute nodes to manually set VIF type to binding_failed
node_override_vif_bridge =	(ListOpt) Nova compute nodes to manually set VIF type to bridge
node_override_vif_hostdev =	(ListOpt) Nova compute nodes to manually set VIF type to hostdev
node_override_vif_hyperv =	(ListOpt) Nova compute nodes to manually set VIF type to hyperv
node_override_vif_ivs =	(ListOpt) Nova compute nodes to manually set VIF type to ivs
node_override_vif_midonet =	(ListOpt) Nova compute nodes to manually set VIF type to midonet
node_override_vif_mlnx_direct =	(ListOpt) Nova compute nodes to manually set VIF type to mlnx_direct
node_override_vif_other =	(ListOpt) Nova compute nodes to manually set VIF type to other
node_override_vif_ovs =	(ListOpt) Nova compute nodes to manually set VIF type to ovs
node_override_vif_unbound =	(ListOpt) Nova compute nodes to manually set VIF type to unbound
vif_type = ovs	(StrOpt) Virtual interface type to configure on Nova compute nodes
vif_types = unbound, binding_failed, ovs, ivs, bridge, 802.1qbg, 802.1qbh, hyperv, midonet, mlnx_direct, hostdev, other	(ListOpt) List of allowed vif_type values.
[RESTPROXY]	
add_meta_server_route = True	(BoolOpt) Flag to decide if a route to the metadata server should be injected into the VM
auto_sync_on_failure = True	(BoolOpt) If neutron fails to create a resource because the backend controller doesn't know of a dependency, the plugin automatically triggers a full data synchronization to the controller.
cache_connections = True	(BoolOpt) Re-use HTTP/HTTPS connections to the controller.
consistency_interval = 60	(IntOpt) Time between verifications that the backend controller database is consistent with Neutron
neutron_id = neutron-oslo	(StrOpt) User defined identifier for this Neutron deployment
no_ssl_validation = False	(BoolOpt) Disables SSL certificate validation for controllers
server_auth = None	(StrOpt) The username and password for authenticating against the Big Switch or Floodlight controller.

Configuration option = Default value	Description
server_ssl = True	(BoolOpt) If True, Use SSL when connecting to the Big Switch or Floodlight controller.
server_timeout = 10	(IntOpt) Maximum number of seconds to wait for proxy request to connect and complete.
servers = localhost:8800	(ListOpt) A comma separated list of Big Switch or Floodlight servers and port numbers. The plugin proxies the requests to the Big Switch/Floodlight server, which performs the networking configuration. Only onserver is needed per deployment, but you may wish to deploy multiple servers to support failover.
ssl_cert_directory = /etc/neutron/plugins/bigswitch/ssl	(StrOpt) Directory containing ca_certs and host_certs certificate directories.
ssl_sticky = True	(BoolOpt) Trust and store the first certificate received for each controller address and use it to validate future connections to that address.
sync_data = False	(BoolOpt) Sync data on connect
thread_pool_size = 4	(IntOpt) Maximum number of threads to spawn to handle large volumes of port creations.
[RESTPROXYAGENT]	
integration_bridge = br-int	(StrOpt) Name of integration bridge on compute nodes used for security group insertion.
polling_interval = 5	(IntOpt) Seconds between agent checks for port changes
virtual_switch_type = ovs	(StrOpt) Virtual switch type.
[ROUTER]	
max_router_rules = 200	(IntOpt) Maximum number of router rules
tenant_default_router_rule = ['*:any:any:permit']	(MultiStrOpt) The default router rules installed in new tenant routers. Repeat the config option for each rule. Format is <tenant>:<source>:<destination>:<action> Use an * to specify default for all tenants.

Brocade configuration options

Table 7.3. Description of configuration options for brocade

Configuration option = Default value	Description
[PHYSICAL_INTERFACE]	
physical_interface = eth0	(StrOpt) The network interface to use when creating a port
[SWITCH]	
address =	(StrOpt) The address of the host to SSH to
ostype = NOS	(StrOpt) Currently unused
password =	(StrOpt) The SSH password to use
username =	(StrOpt) The SSH username to use

CISCO configuration options

Table 7.4. Description of configuration options for cisco

Configuration option = Default value	Description
[CISCO]	
model_class = neutron.plugins.cisco.models.virt_phy_sw_v2.VirtualPhysicalSwitchModelV2	(StrOpt) Model Class

Configuration option = Default value	Description
nexus_driver = neutron.plugins.cisco.test.nexus.fake_nexus_driver.CiscoNEXUSFakeDriver	(StrOpt) Nexus Driver Name
nexus_l3_enable = False	(BoolOpt) Enable L3 support on the Nexus switches
provider_vlan_auto_create = True	(BoolOpt) Provider VLANs are automatically created as needed on the Nexus switch
provider_vlan_auto_trunk = True	(BoolOpt) Provider VLANs are automatically trunked as needed on the ports of the Nexus switch
provider_vlan_name_prefix = p-	(StrOpt) VLAN Name prefix for provider vlans
svi_round_robin = False	(BoolOpt) Distribute SVI interfaces over all switches
vlan_name_prefix = q-	(StrOpt) VLAN Name prefix
[CISCO_N1K]	
bridge_mappings =	(StrOpt) N1K Bridge Mappings
default_network_profile = default_network_profile	(StrOpt) N1K default network profile
default_policy_profile = service_profile	(StrOpt) N1K default policy profile
enable_tunneling = True	(BoolOpt) N1K Enable Tunneling
integration_bridge = br-int	(StrOpt) N1K Integration Bridge
network_node_policy_profile = dhcp_pp	(StrOpt) N1K policy profile for network node
network_vlan_ranges = vlan:1:4095	(StrOpt) N1K Network VLAN Ranges
poll_duration = 10	(StrOpt) N1K Policy profile polling duration in seconds
tenant_network_type = local	(StrOpt) N1K Tenant Network Type
tunnel_bridge = br-tun	(StrOpt) N1K Tunnel Bridge
vxlan_id_ranges = 5000:10000	(StrOpt) N1K VXLAN ID Ranges
[CISCO_PLUGINS]	
nexus_plugin = neutron.plugins.cisco.nexus.cisco_nexus_plugin_v2.NexusPlugin	(StrOpt) Nexus Switch to use
vswitch_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2	(StrOpt) Virtual Switch to use
[cisco_csr_ipsec]	
status_check_interval = 60	(IntOpt) Status check interval for Cisco CSR IPsec connections
[ml2_cisco]	
svi_round_robin = False	(BoolOpt) Distribute SVI interfaces over all switches
vlan_name_prefix = q-	(StrOpt) VLAN Name prefix

CloudBase Hyper-V Agent configuration options

Table 7.5. Description of configuration options for hyperv_agent

Configuration option = Default value	Description
[AGENT]	
enable_metrics_collection = False	(BoolOpt) Enables metrics collections for switch ports by using Hyper-V's metric APIs. Collected data can be retrieved by other apps and services, e.g.: Ceilometer. Requires Hyper-V / Windows Server 2012 and above
local_network_vswitch = private	(StrOpt) Private vswitch name used for local networks
metrics_max_retries = 100	(IntOpt) Specifies the maximum number of retries to enable Hyper-V's port metrics collection. The agent will try to enable the feature once every polling_interval period for at most metrics_max_retries or until it succeeds.

Configuration option = Default value	Description
physical_network_vswitch_mappings =	(ListOpt) List of <physical_network>:<vswitch> where the physical networks can be expressed with wildcards, e.g.: "*" :external"
polling_interval = 2	(IntOpt) The number of seconds the agent will wait between polling for local device changes.
[HYPERV]	
network_vlan_ranges =	(ListOpt) List of <physical_network>:<vlan_min>:<vlan_max> or <physical_network>
tenant_network_type = local	(StrOpt) Network type for tenant networks (local, flat, vlan or none)
[hyperv]	
force_hyperv_utils_v1 = False	(BoolOpt) Force V1 WMI utility classes

Embrane configuration options

Table 7.6. Description of configuration options for embrane

Configuration option = Default value	Description
[heleos]	
admin_username = admin	(StrOpt) ESM admin username.
async_requests = True	(BoolOpt) Define if the requests have run asynchronously or not
dummy_utif_id = None	(StrOpt) Dummy user traffic Security Zone id
esm_mgmt = None	(StrOpt) ESM management root address
inband_id = None	(StrOpt) In band Security Zone id
mgmt_id = None	(StrOpt) Management Security Zone id
oob_id = None	(StrOpt) Out of band Security Zone id
resource_pool_id = default	(StrOpt) Shared resource pool id
router_image = None	(StrOpt) Router image id (Embrane FW/VPN)

IBM SDN-VE configuration options

Table 7.7. Description of configuration options for sdnve

Configuration option = Default value	Description
[SDNVE]	
base_url = /one/nb/v2/	(StrOpt) Base URL for SDN-VE controller REST API
controller_ips = 127.0.0.1	(ListOpt) List of IP addresses of SDN-VE controller(s)
default_tenant_type = OF	(StrOpt) Tenant type: OF (default) and OVERLAY
format = json	(StrOpt) SDN-VE request/response format
info = sdnve_info_string	(StrOpt) SDN-VE RPC subject
integration_bridge = None	(StrOpt) Integration bridge to use
interface_mappings =	(ListOpt) List of <physical_network_name>:<interface_name>
of_signature = SDNVE-OF	(StrOpt) The string in tenant description that indicates the tenant is a OF tenant
out_of_band = True	(BoolOpt) Indicating if controller is out of band or not
overlay_signature = SDNVE-OVERLAY	(StrOpt) The string in tenant description that indicates the tenant is a OVERLAY tenant

Configuration option = Default value	Description
password = admin	(StrOpt) SDN-VE administrator password
port = 8443	(StrOpt) SDN-VE controller port number
reset_bridge = True	(BoolOpt) Reset the integration bridge before use
use_fake_controller = False	(BoolOpt) If set to True uses a fake controller.
userid = admin	(StrOpt) SDN-VE administrator user id
[SDNVE_AGENT]	
polling_interval = 2	(IntOpt) Agent polling interval if necessary
root_helper = sudo	(StrOpt) Using root helper
rpc = True	(BoolOpt) Whether using rpc

Linux bridge Agent configuration options

Table 7.8. Description of configuration options for linuxbridge_agent

Configuration option = Default value	Description
[LINUX_BRIDGE]	
physical_interface_mappings =	(ListOpt) List of <physical_network>:<physical_interface>
[VLANS]	
network_vlan_ranges =	(ListOpt) List of <physical_network>:<vlan_min>:<vlan_max> or <physical_network>
tenant_network_type = local	(StrOpt) Network type for tenant networks (local, vlan, or none)
[VXLAN]	
enable_vxlan = False	(BoolOpt) Enable VXLAN on the agent. Can be enabled when agent is managed by ml2 plugin using linuxbridge mechanism driver
l2_population = False	(BoolOpt) Extension to use alongside ml2 plugin's l2population mechanism driver. It enables the plugin to populate VXLAN forwarding table.
local_ip =	(StrOpt) Local IP address of the VXLAN endpoints.
tos = None	(IntOpt) TOS for vxlan interface protocol packets.
ttl = None	(IntOpt) TTL for vxlan interface protocol packets.
vxlan_group = 224.0.0.1	(StrOpt) Multicast group for vxlan interface.

Mellanox configuration options

Table 7.9. Description of configuration options for mlnx

Configuration option = Default value	Description
[ESWITCH]	
backoff_rate = 2	(IntOpt) backoff rate multiplier for waiting period between retries for request to daemon, i.e. value of 2 will double the request timeout each retry
daemon_endpoint = tcp://127.0.0.1:60001	(StrOpt) eswitch daemon end point
physical_interface_mappings =	(ListOpt) List of <physical_network>:<physical_interface>
request_timeout = 3000	(IntOpt) The number of milliseconds the agent will wait for response on request to daemon.
[MLNX]	

Configuration option = Default value	Description
network_vlan_ranges = default:1:1000	(ListOpt) List of <physical_network>:<vlan_min>:<vlan_max> or <physical_network>
physical_network_type = eth	(StrOpt) Physical network type for provider network (eth or ib)
physical_network_type_mappings =	(ListOpt) List of <physical_network>:<physical_network_type> with physical_network_type is either eth or ib
tenant_network_type = vlan	(StrOpt) Network type for tenant networks (local, vlan, or none)

Meta Plug-in configuration options

The Meta Plug-in allows you to use multiple plug-ins at the same time.

Table 7.10. Description of configuration options for meta

Configuration option = Default value	Description
[META]	
default_flavor =	(StrOpt) Default flavor to use
default_l3_flavor =	(StrOpt) Default L3 flavor to use
extension_map =	(StrOpt) A list of extensions, per plugin, to load.
l3_plugin_list =	(StrOpt) List of L3 plugins to load
plugin_list =	(StrOpt) List of plugins to load
rpc_flavor =	(StrOpt) Flavor of which plugin handles RPC
supported_extension_aliases =	(StrOpt) Supported extension aliases

Modular Layer 2 (ml2) configuration options

The Modular Layer 2 (ml2) plug-in has two components: network types and mechanisms. You can configure these components separately. This section describes these configuration options.



MTU bug with VXLAN tunnelling

Due to a bug in Linux Bridge software maximum transmission unit (MTU) handling, using VXLAN tunnels does not work by default.

- A simple workaround is to increase the MTU value of the physical interface and physical switch fabric by at least 50 bytes. For example, increase the MTU value to 1550. This value enables an automatic 50-byte MTU difference between the physical interface (1500) and the VXLAN interface (automatically 1500-50 = 1450). An MTU value of 1450 causes issues when virtual machine taps are configured at an MTU value of 1500.
- Another workaround is to decrease the virtual ethernet devices' MTU. Set the `network_device_mtu` option to 1450 in the `neutron.conf` file, and set all guest virtual machines' MTU to the same value by using a DHCP option. For information about how to use this option, see [Configure OVS plug-in](#).

Table 7.11. Description of configuration options for ml2

Configuration option = Default value	Description
[ml2]	
mechanism_drivers =	(ListOpt) An ordered list of networking mechanism driver entrypoints to be loaded from the neutron.ml2.mechanism_drivers namespace.
tenant_network_types = local	(ListOpt) Ordered list of network_types to allocate as tenant networks.
type_drivers = local, flat, vlan, gre, vxlan	(ListOpt) List of network type driver entrypoints to be loaded from the neutron.ml2.type_drivers namespace.

Modular Layer 2 (ml2) Flat Type configuration options**Table 7.12. Description of configuration options for ml2_flat**

Configuration option = Default value	Description
[ml2_type_flat]	
flat_networks =	(ListOpt) List of physical_network names with which flat networks can be created. Use * to allow flat networks with arbitrary physical_network names.

Modular Layer 2 (ml2) GRE Type configuration options**Table 7.13. Description of configuration options for ml2_gre**

Configuration option = Default value	Description
[ml2_type_gre]	
tunnel_id_ranges =	(ListOpt) Comma-separated list of <tun_min>:<tun_max> tuples enumerating ranges of GRE tunnel IDs that are available for tenant network allocation

Modular Layer 2 (ml2) VLAN Type configuration options**Table 7.14. Description of configuration options for ml2_vlan**

Configuration option = Default value	Description
[ml2_type_vlan]	
network_vlan_ranges =	(ListOpt) List of <physical_network>:<vlan_min>:<vlan_max> or <physical_network> specifying physical_network names usable for VLAN provider and tenant networks, as well as ranges of VLAN tags on each available for allocation to tenant networks.

Modular Layer 2 (ml2) VXLAN Type configuration options**Table 7.15. Description of configuration options for ml2_vxlan**

Configuration option = Default value	Description
[ml2_type_vxlan]	
vni_ranges =	(ListOpt) Comma-separated list of <vni_min>:<vni_max> tuples enumerating ranges of VXLAN VNI IDs that are available for tenant network allocation
vxlan_group = None	(StrOpt) Multicast group for VXLAN. If unset, disables VXLAN multicast mode.

Modular Layer 2 (ml2) Arista Mechanism configuration options**Table 7.16. Description of configuration options for ml2_arista**

Configuration option = Default value	Description
[ml2_arista]	
eapi_host =	(StrOpt) Arista EOS IP address. This is required field.If not set, all communications to Arista EOSwill fail
eapi_password =	(StrOpt) Password for Arista EOS. This is required field.if not set, all communications to Arista EOSwill fail
eapi_username =	(StrOpt) Username for Arista EOS. This is required field.if not set, all communications to Arista EOSwill fail
region_name = RegionOne	(StrOpt) Defines Region Name that is assigned to this OpenStackController. This is useful when multipleOpenStack/Neutron controllers are managing the sameArista HW clusters. Note that this name must match withthe region name registered (or known) to keystoneservice. Authentication with Keysothne is performed byEOS. This is optional. If not set, a value of"RegionOne" is assumed
sync_interval = 180	(IntOpt) Sync interval in seconds between Neutron plugin andEOS. This interval defines how often thesynchronization is performed. This is an optionalfield. If not set, a value of 180 seconds is assumed
use_fqdn = True	(BoolOpt) Defines if hostnames are sent to Arista EOS as FQDNs("node1.domain.com") or as short names ("node1").This is optional. If not set, a value of "True"is assumed.

Modular Layer 2 (ml2) BigSwitch Mechanism configuration options**Table 7.17. Description of configuration options for ml2_bigswitch**

Configuration option = Default value	Description
[NOVA]	
node_override_vif_802.1qbg =	(ListOpt) Nova compute nodes to manually set VIF type to 802.1qbg
node_override_vif_802.1qbh =	(ListOpt) Nova compute nodes to manually set VIF type to 802.1qbh
node_override_vif_binding_failed =	(ListOpt) Nova compute nodes to manually set VIF type to binding_failed
node_override_vif_bridge =	(ListOpt) Nova compute nodes to manually set VIF type to bridge
node_override_vif_hostdev =	(ListOpt) Nova compute nodes to manually set VIF type to hostdev
node_override_vif_hyperv =	(ListOpt) Nova compute nodes to manually set VIF type to hyperv
node_override_vif_ivs =	(ListOpt) Nova compute nodes to manually set VIF type to ivs
node_override_vif_midonet =	(ListOpt) Nova compute nodes to manually set VIF type to midonet
node_override_vif_mlnx_direct =	(ListOpt) Nova compute nodes to manually set VIF type to mlnx_direct
node_override_vif_other =	(ListOpt) Nova compute nodes to manually set VIF type to other

Configuration option = Default value	Description
node_override_vif_ovs =	(ListOpt) Nova compute nodes to manually set VIF type to ovs
node_override_vif_unbound =	(ListOpt) Nova compute nodes to manually set VIF type to unbound
vif_type = ovs	(StrOpt) Virtual interface type to configure on Nova compute nodes
vif_types = unbound, binding_failed, ovs, ivs, bridge, 802.1qbg, 802.1qbh, hyperv, midonet, mlnx_direct, hostdev, other	(ListOpt) List of allowed vif_type values.
[RESTPROXY]	
add_meta_server_route = True	(BoolOpt) Flag to decide if a route to the metadata server should be injected into the VM
auto_sync_on_failure = True	(BoolOpt) If neutron fails to create a resource because the backend controller doesn't know of a dependency, the plugin automatically triggers a full data synchronization to the controller.
cache_connections = True	(BoolOpt) Re-use HTTP/HTTPS connections to the controller.
consistency_interval = 60	(IntOpt) Time between verifications that the backend controller database is consistent with Neutron
neutron_id = neutron-oslo	(StrOpt) User defined identifier for this Neutron deployment
no_ssl_validation = False	(BoolOpt) Disables SSL certificate validation for controllers
server_auth = None	(StrOpt) The username and password for authenticating against the Big Switch or Floodlight controller.
server_ssl = True	(BoolOpt) If True, Use SSL when connecting to the Big Switch or Floodlight controller.
server_timeout = 10	(IntOpt) Maximum number of seconds to wait for proxy request to connect and complete.
servers = localhost:8800	(ListOpt) A comma separated list of Big Switch or Floodlight servers and port numbers. The plugin proxies the requests to the Big Switch/Floodlight server, which performs the networking configuration. Only onserver is needed per deployment, but you may wish to deploy multiple servers to support failover.
ssl_cert_directory = /etc/neutron/plugins/bigswitch/ssl	(StrOpt) Directory containing ca_certs and host_certs certificate directories.
ssl_sticky = True	(BoolOpt) Trust and store the first certificate received for each controller address and use it to validate future connections to that address.
sync_data = False	(BoolOpt) Sync data on connect
thread_pool_size = 4	(IntOpt) Maximum number of threads to spawn to handle large volumes of port creations.
[RESTPROXYAGENT]	
integration_bridge = br-int	(StrOpt) Name of integration bridge on compute nodes used for security group insertion.
polling_interval = 5	(IntOpt) Seconds between agent checks for port changes
virtual_switch_type = ovs	(StrOpt) Virtual switch type.
[ROUTER]	
max_router_rules = 200	(IntOpt) Maximum number of router rules
tenant_default_router_rule = ['*:any:any:permit']	(MultiStrOpt) The default router rules installed in new tenant routers. Repeat the config option for each rule.

Configuration option = Default value	Description
	Format is <tenant>:<source>:<destination>:<action> Use an * to specify default for all tenants.

Modular Layer 2 (ml2) Brocade Mechanism configuration options

Table 7.18. Description of configuration options for ml2_brocade

Configuration option = Default value	Description
[ml2_brocade]	
address =	(StrOpt) The address of the host to SSH to
ostype = NOS	(StrOpt) Unused
password = password	(StrOpt) The SSH password to use
physical_networks =	(StrOpt) Allowed physical networks
username = admin	(StrOpt) The SSH username to use

Modular Layer 2 (ml2) Cisco Mechanism configuration options

Table 7.19. Description of configuration options for ml2_cisco

Configuration option = Default value	Description
[ml2_cisco]	
managed_physical_network = None	(StrOpt) The physical network managed by the switches.

Modular Layer 2 (ml2) Mellanox Mechanism configuration options

Table 7.20. Description of configuration options for ml2_mlnx

Configuration option = Default value	Description
[ESWITCH]	
apply_profile_patch = False	(BoolOpt) Enable server compatibility with old nova
vnic_type = mlnx_direct	(StrOpt) Type of VM network interface: mlnx_direct or hostdev

Modular Layer 2 (ml2) OpenDaylight Mechanism configuration options

Table 7.21. Description of configuration options for ml2_odl

Configuration option = Default value	Description
[ml2_odl]	
password = None	(StrOpt) HTTP password for authentication
session_timeout = 30	(IntOpt) Tomcat session timeout in minutes.
timeout = 10	(IntOpt) HTTP timeout in seconds.
url = None	(StrOpt) HTTP URL of OpenDaylight REST interface.
username = None	(StrOpt) HTTP username for authentication

Modular Layer 2 (ml2) OpenFlow Mechanism configuration options

Table 7.22. Description of configuration options for ml2_ofa

Configuration option = Default value	Description
[DEFAULT]	
ofp_listen_host =	(StrOpt) openflow listen host

Configuration option = Default value	Description
ofp_ssl_listen_port = 6633	(IntOpt) openflow ssl listen port
ofp_tcp_listen_port = 6633	(IntOpt) openflow tcp listen port
[AGENT]	
get_datapath_retry_times = 60	(IntOpt) Number of seconds to retry acquiring an Open vSwitch datapath

Modular Layer 2 (ml2) L2 Population Mechanism configuration options

Table 7.23. Description of configuration options for ml2_l2pop

Configuration option = Default value	Description
[l2pop]	
agent_boot_time = 180	(IntOpt) Delay within which agent is expected to update existing ports when it restarts

Modular Layer 2 (ml2) Tail-f NCS Mechanism configuration options

Table 7.24. Description of configuration options for ml2_ncs

Configuration option = Default value	Description
[ml2_ncs]	
password = None	(StrOpt) HTTP password for authentication
timeout = 10	(IntOpt) HTTP timeout in seconds.
url = None	(StrOpt) HTTP URL of Tail-f NCS REST interface.
username = None	(StrOpt) HTTP username for authentication

MidoNet configuration options

Table 7.25. Description of configuration options for midonet

Configuration option = Default value	Description
[MIDONET]	
midonet_host_uuid_path = /etc/midolman/host_uuid.properties	(StrOpt) Path to midonet host uuid file
midonet_uri = http://localhost:8080/midonet-api	(StrOpt) MidoNet API server URI.
mode = dev	(StrOpt) Operational mode. Internal dev use only.
password = passw0rd	(StrOpt) MidoNet admin password.
project_id = 77777777-7777-7777-7777-777777777777	(StrOpt) ID of the project that MidoNet admin user belongs to.
provider_router_id = None	(StrOpt) Virtual provider router ID.
username = admin	(StrOpt) MidoNet admin username.

NEC configuration options

Table 7.26. Description of configuration options for nec

Configuration option = Default value	Description
[OFC]	
api_max_attempts = 3	(IntOpt) Maximum attempts per OFC API request. NEC plugin retries API request to OFC when OFC returns ServiceUnavailable (503). The value must be greater than 0.

Configuration option = Default value	Description
cert_file = None	(StrOpt) Certificate file
driver = tremas	(StrOpt) Driver to use
enable_packet_filter = True	(BoolOpt) Enable packet filter
host = 127.0.0.1	(StrOpt) Host to connect to
insecure_ssl = False	(BoolOpt) Disable SSL certificate verification
key_file = None	(StrOpt) Key file
path_prefix =	(StrOpt) Base URL of OFC REST API. It is prepended to each API request.
port = 8888	(StrOpt) Port to connect to
use_ssl = False	(BoolOpt) Use SSL to connect
[PROVIDER]	
default_router_provider = l3-agent	(StrOpt) Default router provider to use.
router_providers = l3-agent, openflow	(ListOpt) List of enabled router providers.
[fwaas]	
driver =	(StrOpt) Name of the FWaaS Driver

Nuage configuration options

Table 7.27. Description of configuration options for nuage

Configuration option = Default value	Description
[RESTPROXY]	
auth_resource =	(StrOpt) Nuage provided uri for initial authorization to access VSD
base_uri = /	(StrOpt) Nuage provided base uri to reach out to VSD
default_floatingip_quota = 254	(IntOpt) Per Net Partition quota of floating ips
default_net_partition_name = OpenStackDefaultNetPartition	(StrOpt) Default Network partition in which VSD will orchestrate network resources using openstack
organization = system	(StrOpt) Organization name in which VSD will orchestrate network resources using openstack
server = localhost:8800	(StrOpt) IP Address and Port of Nuage's VSD server
serverauth = username:password	(StrOpt) Username and password for authentication
serverssl = False	(BoolOpt) Boolean for SSL connection with VSD server

One Convergence NVSD configuration options

Table 7.28. Description of configuration options for nvsd

Configuration option = Default value	Description
[AGENT]	
integration_bridge = br-int	(StrOpt) integration bridge
[nvsd]	
nvsd_ip = 127.0.0.1	(StrOpt) NVSD Controller IP address
nvsd_passwd = oc123	(StrOpt) NVSD Controller password
nvsd_port = 8082	(IntOpt) NVSD Controller Port number
nvsd_retries = 0	(IntOpt) Number of login retries to NVSD controller
nvsd_user = ocplugin	(StrOpt) NVSD Controller username

Configuration option = Default value	Description
request_timeout = 30	(IntOpt) NVSD controller REST API request timeout in seconds

VMware NSX configuration options

Table 7.29. Description of configuration options for vmware

Configuration option = Default value	Description
[DEFAULT]	
default_interface_name = breth0	(StrOpt) Name of the interface on a L2 Gateway transport node which should be used by default when setting up a network connection
default_l2_gw_service_uuid = None	(StrOpt) Unique identifier of the NSX L2 Gateway service which will be used by default for network gateways
default_l3_gw_service_uuid = None	(StrOpt) Unique identifier of the NSX L3 Gateway service which will be used for implementing routers and floating IPs
default_service_cluster_uuid = None	(StrOpt) Unique identifier of the Service Cluster which will be used by logical services like dhcp and metadata
default_tz_uuid = None	(StrOpt) This is uuid of the default NSX Transport zone that will be used for creating tunneled isolated "Neutron" networks. It needs to be created in NSX before starting Neutron with the nsx plugin.
http_timeout = 10	(IntOpt) Time before aborting a request
nsx_controllers = None	(ListOpt) Lists the NSX controllers in this cluster
nsx_password = admin	(StrOpt) Password for NSX controllers in this cluster
nsx_user = admin	(StrOpt) User name for NSX controllers in this cluster
redirects = 2	(IntOpt) Number of times a redirect should be followed
req_timeout = 30	(IntOpt) Total time limit for a cluster request
retries = 2	(IntOpt) Number of time a request should be retried
[ESWITCH]	
retries = 3	(IntOpt) The number of retries the agent will send request to daemon before giving up
[NSX]	
agent_mode = agent	(StrOpt) The mode used to implement DHCP/metadata services.
concurrent_connections = 10	(IntOpt) Maximum concurrent connections to each NSX controller.
default_transport_type = stt	(StrOpt) The default network transport type to use (stt, gre, bridge, ipsec_gre, or ipsec_stt)
max_ip_per_bridged_ls = 5000	(IntOpt) Maximum number of ports of a logical switch on a bridged transport zone (default 5000)
max_ip_per_overlay_ls = 256	(IntOpt) Maximum number of ports of a logical switch on an overlay transport zone (default 256)
metadata_mode = access_network	(StrOpt) If set to access_network this enables a dedicated connection to the metadata proxy for metadata server access via Neutron router. If set to dhcp_host_route this enables host route injection via the dhcp agent. This option is only useful if running on a host that does not support namespaces otherwise access_network should be used.
nsx_gen_timeout = -1	(IntOpt) Number of seconds a generation id should be valid for (default -1 meaning do not time out)

Configuration option = Default value	Description
replication_mode = service	(StrOpt) The default option leverages service nodes to perform packet replication though one could set to this to 'source' to perform replication locally. This is useful if one does not want to deploy a service node(s).
[NSX_DHCP]	
default_lease_time = 43200	(IntOpt) Default DHCP lease time
domain_name = openstacklocal	(StrOpt) Domain to use for building the hostnames
extra_domain_name_servers =	(ListOpt) Comma separated list of additional domain name servers
[NSX_LSN]	
sync_on_missing_data = False	(BoolOpt) Pull LSN information from NSX in case it is missing from the local data store. This is useful to rebuild the local store in case of server recovery.
[NSX_METADATA]	
metadata_server_address = 127.0.0.1	(StrOpt) IP address used by Metadata server.
metadata_server_port = 8775	(IntOpt) TCP Port used by Metadata server.
metadata_shared_secret =	(StrOpt) Shared secret to sign instance-id request
[NSX_SYNC]	
always_read_status = False	(BoolOpt) Always read operational status from backend on show operations. Enabling this option might slow down the system.
max_random_sync_delay = 0	(IntOpt) Maximum value for the additional random delay in seconds between runs of the state synchronization task
min_chunk_size = 500	(IntOpt) Minimum number of resources to be retrieved from NSX during state synchronization
min_sync_req_delay = 1	(IntOpt) Minimum delay, in seconds, between two state synchronization queries to NSX. It must not exceed state_sync_interval
state_sync_interval = 10	(IntOpt) Interval in seconds between runs of the state synchronization task. Set it to 0 to disable it
[vcns]	
datacenter_moid = None	(StrOpt) Optional parameter identifying the ID of datacenter to deploy NSX Edges
datastore_id = None	(StrOpt) Optional parameter identifying the ID of datastore to deploy NSX Edges
deployment_container_id = None	(StrOpt) Optional parameter identifying the ID of datastore to deploy NSX Edges
external_network = None	(StrOpt) Network ID for physical network connectivity
manager_uri = None	(StrOpt) uri for vsm
password = default	(StrOpt) Password for vsm
resource_pool_id = None	(StrOpt) Optional parameter identifying the ID of resource to deploy NSX Edges
task_status_check_interval = 2000	(IntOpt) Task status check interval
user = admin	(StrOpt) User name for vsm

Open vSwitch Agent configuration options

Table 7.30. Description of configuration options for openvswitch_agent

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
ovs_integration_bridge = br-int	(StrOpt) Name of Open vSwitch bridge to use
ovs_use_veth = False	(BoolOpt) Uses veth for an interface or not
ovs_vsctl_timeout = 10	(IntOpt) Timeout in seconds for ovs-vsctl commands
[AGENT]	
l2_population = False	(BoolOpt) Use ml2 l2population mechanism driver to learn remote mac and IPs and improve tunnel scalability
minimize_polling = True	(BoolOpt) Minimize polling by monitoring ovssdb for interface changes.
ovssdb_monitor_respawn_interval = 30	(IntOpt) The number of seconds to wait before respawning the ovssdb monitor after losing communication with it
tunnel_types =	(ListOpt) Network types supported by the agent (gre and/or vxlan)
veth_mtu = None	(IntOpt) MTU size of veth interfaces
vxlan_udp_port = 4789	(IntOpt) The UDP port to use for VXLAN tunnels.
[CISCO_N1K]	
local_ip = 10.0.0.3	(StrOpt) N1K Local IP
[OVS]	
bridge_mappings =	(ListOpt) List of <physical_network>:<bridge>
enable_tunneling = False	(BoolOpt) Enable tunneling support
int_peer_patch_port = patch-tun	(StrOpt) Peer patch port in integration bridge for tunnel bridge
integration_bridge = br-int	(StrOpt) Integration bridge to use
local_ip =	(StrOpt) Local IP address of GRE tunnel endpoints.
network_vlan_ranges =	(ListOpt) List of <physical_network>:<vlan_min>:<vlan_max> or <physical_network>
tenant_network_type = local	(StrOpt) Network type for tenant networks (local, vlan, gre, vxlan, or none)
tun_peer_patch_port = patch-int	(StrOpt) Peer patch port in tunnel bridge for integration bridge
tunnel_bridge = br-tun	(StrOpt) Tunnel bridge to use
tunnel_id_ranges =	(ListOpt) List of <tun_min>:<tun_max>
tunnel_type =	(StrOpt) The type of tunnels to use when utilizing tunnels, either 'gre' or 'vxlan'

PLUMgrid configuration options

Table 7.31. Description of configuration options for plumgrid

Configuration option = Default value	Description
[plumgriddirector]	
director_server = localhost	(StrOpt) PLUMgrid Director server to connect to
director_server_port = 8080	(StrOpt) PLUMgrid Director server port to connect to
password = password	(StrOpt) PLUMgrid Director admin password
servertimeout = 5	(IntOpt) PLUMgrid Director server timeout
username = username	(StrOpt) PLUMgrid Director admin username

Ryu configuration options

Table 7.32. Description of configuration options for ryu

Configuration option = Default value	Description
[OVS]	
openflow_rest_api = 127.0.0.1:8080	(StrOpt) OpenFlow REST API location
ovsdb_interface = None	(StrOpt) OVSDDB interface to connect to
ovsdb_ip = None	(StrOpt) OVSDDB IP to connect to
ovsdb_port = 6634	(IntOpt) OVSDDB port to connect to
tunnel_interface = None	(StrOpt) Tunnel interface to use
tunnel_ip = None	(StrOpt) Tunnel IP to use
tunnel_key_max = 16777215	(IntOpt) Maximum tunnel ID to use
tunnel_key_min = 1	(IntOpt) Minimum tunnel ID to use

Configure the Oslo RPC messaging system

OpenStack projects use an open standard for messaging middleware known as AMQP. This messaging middleware enables the OpenStack services that run on multiple servers to talk to each other. OpenStack Oslo RPC supports three implementations of AMQP: RabbitMQ, Qpid, and ZeroMQ.

Configure RabbitMQ

OpenStack Oslo RPC uses RabbitMQ by default. Use these options to configure the RabbitMQ message system. The `rpc_backend` option is optional as long as RabbitMQ is the default messaging system. However, if it is included the configuration, you must set it to `neutron.openstack.common.rpc.impl_kombu`.

```
rpc_backend=neutron.openstack.common.rpc.impl_kombu
```

Use these options to configure the RabbitMQ messaging system. You can configure messaging communication for different installation scenarios, tune retries for RabbitMQ, and define the size of the RPC thread pool. To monitor notifications through RabbitMQ, you must set the `notification_driver` option to `neutron.notifier.rabbit_notifier` in the `neutron.conf` file:

Table 7.33. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count)
rabbit_password = guest	(StrOpt) The RabbitMQ password

Configuration option = Default value	Description
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ
rabbit_userid = guest	(StrOpt) The RabbitMQ userid
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host

Table 7.34. Description of configuration options for kombu

Configuration option = Default value	Description
[DEFAULT]	
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled)
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled)
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled)
kombu_ssl_version =	(StrOpt) If SSL is enabled, the SSL version to use. Valid values are TLSv1, SSLv23 and SSLv3. SSLv2 might be available on some distributions.

Configure Qpid

Use these options to configure the Qpid messaging system for OpenStack Oslo RPC. Qpid is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `neutron.conf` file:

```
rpc_backend=neutron.openstack.common.rpc.impl_qpid
```

This critical option points the compute nodes to the Qpid broker (server). Set the `qpid_hostname` option to the host name where the broker runs in the `neutron.conf` file.



Note

The `--qpid_hostname` option accepts a host name or IP address value.

```
qpid_hostname=hostname.example.com
```

If the Qpid broker listens on a port other than the AMQP default of 5672, you must set the `qpid_port` option to that value:

```
qpid_port=12345
```

If you configure the Qpid broker to require authentication, you must add a user name and password to the configuration:

```
qpid_username=username
qpid_password=password
```

By default, TCP is used as the transport. To enable SSL, set the `qpid_protocol` option:

```
qpid_protocol=ssl
```

Use these additional options to configure the Qpid messaging driver for OpenStack Oslo RPC. These options are used infrequently.

Table 7.35. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats
qpid_hostname = localhost	(StrOpt) Qpid broker hostname
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs
qpid_password =	(StrOpt) Password for qpid connection
qpid_port = 5672	(IntOpt) Qpid broker port
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth
qpid_tcp_nodelay = True	(BoolOpt) Disable Nagle algorithm
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for qpid connection

Configure ZeroMQ

Use these options to configure the ZeroMQ messaging system for OpenStack Oslo RPC. ZeroMQ is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `neutron.conf` file:

Table 7.36. Description of configuration options for zeromq

Configuration option = Default value	Description
[DEFAULT]	
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets

Configuration option = Default value	Description
rpc_zmq_matchmaker = neutron.openstack.common.rpc.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Configure messaging

Use these common options to configure the RabbitMQ, Qpid, and ZeroMq messaging drivers:

Table 7.37. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
control_exchange = neutron	(StrOpt) AMQP exchange to connect to if using RabbitMQ or Qpid
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
rpc_backend = neutron.openstack.common.rpc.impl_kombu	(StrOpt) The messaging module to use, defaults to kombu.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from call or multical
rpc_thread_pool_size = 64	(IntOpt) Size of RPC thread pool
rpc_workers = 0	(IntOpt) Number of RPC worker processes for service
[AGENT]	
rpc_support_old_agents = False	(BoolOpt) Enable server RPC compatibility with old agents
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON)
[rpc_notifier2]	
topics = notifications	(ListOpt) AMQP topic(s) used for openstack notifications

Table 7.38. Description of configuration options for notifier

Configuration option = Default value	Description
[DEFAULT]	
default_notification_level = INFO	(StrOpt) Default notification level for outgoing notifications
default_publisher_id = \$host	(StrOpt) Default publisher_id for outgoing notifications
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications
notification_topics = notifications	(ListOpt) AMQP topic used for openstack notifications

Table 7.39. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	
host = oslo	(StrOpt) The hostname Neutron is running on
[matchmaker_redis]	
host = 127.0.0.1	(StrOpt) Host to locate redis
password = None	(StrOpt) Password for Redis server. (optional)
port = 6379	(IntOpt) Use this port to connect to redis host.

Agent

Use the following options to alter agent-related settings.

Table 7.40. Description of configuration options for agent

Configuration option = Default value	Description
[DEFAULT]	
external_pids = \$state_path/external/pids	(StrOpt) Location to store child pid files
network_device_mtu = None	(IntOpt) MTU setting for device.

API

Use the following options to alter API-related settings.

Table 7.41. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
allow_bulk = True	(BoolOpt) Allow the usage of the bulk API
allow_pagination = False	(BoolOpt) Allow the usage of the pagination
allow_sorting = False	(BoolOpt) Allow the usage of the sorting
api_extensions_path =	(StrOpt) The path for API extensions
api_paste_config = api-paste.ini	(StrOpt) The API paste config file to use
max_header_line = 16384	(IntOpt) Max header line to accommodate large tokens
max_request_body_size = 114688	(IntOpt) the maximum body size per each request(bytes)
pagination_max_limit = -1	(StrOpt) The maximum number of items returned in a single response, value was 'infinite' or negative integer means no limit
run_external_periodic_tasks = True	(BoolOpt) Some periodic tasks can be run in a separate process. Should we run them here?
service_plugins =	(ListOpt) The service plugins Neutron will use
[service_providers]	
service_provider = []	(MultiStrOpt) Defines providers for advanced services using the format: <service_type>:<name>:<driver>[:default]

Token authentication

Use the following options to alter token authentication settings.

Table 7.42. Description of configuration options for auth_token

Configuration option = Default value	Description
[DEFAULT]	
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) 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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. If MAC, token

Configuration option = Default value	Description
	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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

CADF

Use the following options to alter Cloud Audit Data Federation (CADF) settings.

Table 7.43. Description of configuration options for cadf

Configuration option = Default value	Description
[audit]	
api_audit_map = api_audit_map.conf	(StrOpt) File containing mapping for api paths and service endpoints
namespace = openstack	(StrOpt) namespace prefix for generated id

Compute

Use the following options to alter Compute-related settings.

Table 7.44. Description of configuration options for compute

Configuration option = Default value	Description
[DEFAULT]	
notify_nova_on_port_data_changes = True	(BoolOpt) Send notification to nova when port data (fixed_ips/floatingip) changes so nova can update its cache.
notify_nova_on_port_status_changes = True	(BoolOpt) Send notification to nova when port status changes
nova_admin_auth_url = http://localhost:5000/v2.0	(StrOpt) Authorization URL for connecting to nova in admin context
nova_admin_password = None	(StrOpt) Password for connection to nova in admin context
nova_admin_tenant_id = None	(StrOpt) The uuid of the admin nova tenant
nova_admin_username = None	(StrOpt) Username for connecting to nova in admin context
nova_region_name = None	(StrOpt) Name of nova region to use. Useful if keystone manages more than one region.

Configuration option = Default value	Description
nova_url = http://127.0.0.1:8774/v2	(StrOpt) URL for connection to nova
send_events_interval = 2	(IntOpt) Number of seconds between sending events to nova if there are any events to send.

Database

Use the following options to alter Database-related settings.

Table 7.45. Description of configuration options for db

Configuration option = Default value	Description
[DEFAULT]	
sqlite_db =	(StrOpt) The file name to use with SQLite
sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
connection = sqlite://	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
db_retry_interval = 1	(IntOpt) seconds between db connection retries
idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
max_overflow = 20	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = 10	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
pool_timeout = 10	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
slave_connection =	(StrOpt) The SQLAlchemy connection string used to connect to the slave database
use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost

DHCP agent

Use the following options to alter Database-related settings.

Table 7.46. Description of configuration options for dhcp_agent

Configuration option = Default value	Description
[DEFAULT]	
dnsmasq_config_file =	(StrOpt) Override the default dnsmasq settings with this file
dnsmasq_dns_servers = None	(ListOpt) Comma-separated list of the DNS servers which will be used as forwarders.
dnsmasq_lease_max = 16777216	(IntOpt) Limit number of leases to prevent a denial-of-service.
enable_isolated_metadata = False	(BoolOpt) Support Metadata requests on isolated networks.
enable_metadata_network = False	(BoolOpt) Allows for serving metadata requests from a dedicated network. Requires enable_isolated_metadata = True
num_sync_threads = 4	(IntOpt) Number of threads to use during sync process.
resync_interval = 5	(IntOpt) Interval to resync.
use_namespaces = True	(BoolOpt) Allow overlapping IP.

Embrane LBaaS driver

Use the following options to alter Embrane Loadbalancer-as-a-Service related settings.

Table 7.47. Description of configuration options for embrane_lb

Configuration option = Default value	Description
[heleoslb]	
admin_password = None	(StrOpt) ESM admin password.
admin_username = None	(StrOpt) ESM admin username.
async_requests = None	(BoolOpt) Define if the requests have run asynchronously or not
dummy_utif_id = None	(StrOpt) Dummy user traffic Security Zone id for LBs
esm_mgmt = None	(StrOpt) ESM management root address
inband_id = None	(StrOpt) In band Security Zone id for LBs
lb_flavor = small	(StrOpt) choose LB image flavor to use, accepted values: small, medium
lb_image = None	(StrOpt) Load Balancer image id (Embrane LB)
mgmt_id = None	(StrOpt) Management Security Zone id for LBs
oob_id = None	(StrOpt) Out of band Security Zone id for LBs
resource_pool_id = None	(StrOpt) Shared resource pool id
sync_interval = 60	(IntOpt) resource synchronization interval in seconds

Firewall-as-a-Service driver

Use the following options in the `fwaas_driver.ini` file for the FwaaS driver.

Table 7.48. Description of configuration options for fwaas

Configuration option = Default value	Description
[fwaas]	
enabled = False	(BoolOpt) Enable FWaaS

L3 agent

Use the following options in the `l3_agent.ini` file for the L3 agent.

Table 7.49. Description of configuration options for l3_agent

Configuration option = Default value	Description
[DEFAULT]	
<code>enable_metadata_proxy = True</code>	(BoolOpt) Allow running metadata proxy.
<code>external_network_bridge = br-ex</code>	(StrOpt) Name of bridge used for external network traffic.
<code>gateway_external_network_id =</code>	(StrOpt) UUID of external network for routers implemented by the agents.
<code>handle_internal_only_routers = True</code>	(BoolOpt) Agent should implement routers with no gateway
<code>router_id =</code>	(StrOpt) If namespaces is disabled, the l3 agent can only configure a router that has the matching router ID.
<code>send_arp_for_ha = 0</code>	(IntOpt) Send this many gratuitous ARPs for HA setup, if less than or equal to 0, the feature is disabled

Loadbalancer-as-a-Service agent

Use the following options in the `lbaas_agent.ini` file for the LbaaS agent.

Table 7.50. Description of configuration options for lbaas

Configuration option = Default value	Description
[DEFAULT]	
<code>device_driver =</code> ['neutron.services.loadbalancer.drivers.haproxy.namespace_driver:HAProxyNSDriver']	(MultiStrOpt) Drivers used to manage loadbalancing
<code>loadbalancer_pool_scheduler_driver =</code> neutron.services.loadbalancer.agent_scheduler.ChanceScheduler	(StrOpt) Driver to use for scheduling pool to a default loadbalancer agent
[haproxy]	
<code>loadbalancer_state_path = \$state_path/lbaas</code>	(StrOpt) Location to store config and state files
<code>user_group = nogroup</code>	(StrOpt) The user group
[netscaler_driver]	
<code>netscaler_ncc_password = None</code>	(StrOpt) Password to login to the NetScaler Control Center Server.
<code>netscaler_ncc_uri = None</code>	(StrOpt) The URL to reach the NetScaler Control Center Server.
<code>netscaler_ncc_username = None</code>	(StrOpt) Username to login to the NetScaler Control Center Server.
[radware]	
<code>actions_to_skip = setup_l2_l3</code>	(ListOpt) List of actions that we dont want to push to the completion queue
<code>l2_l3_ctor_params = {'ha_ip_pool_name': 'default', 'allocate_ha_vrrp': True, 'ha_network_name': 'HA-Network', 'service': '_REPLACE_', 'allocate_ha_ips': True}</code>	(DictOpt) l2_l3 workflow constructor params
<code>l2_l3_setup_params = {'data_ip_address': '192.168.200.99', 'data_port': 1, 'gateway': '192.168.200.1', 'ha_port': 2, 'data_ip_mask': '255.255.255.0'}</code>	(DictOpt) l2_l3 workflow setup params
<code>l2_l3_workflow_name = openstack_l2_l3</code>	(StrOpt) l2_l3 workflow name
<code>l4_action_name = BaseCreate</code>	(StrOpt) l4 workflow action name

Configuration option = Default value	Description
l4_workflow_name = openstack_l4	(StrOpt) l4 workflow name
service_adc_type = VA	(StrOpt) Service ADC type
service_adc_version =	(StrOpt) Service ADC version
service_cache = 20	(IntOpt) service cache
service_compression_throughput = 100	(IntOpt) service compression throughput
service_ha_pair = False	(BoolOpt) service HA pair
service_isl_vlan = -1	(IntOpt) A required VLAN for the interswitch link to use
service_resource_pool_ids =	(ListOpt) Resource pool ids
service_session_mirroring_enabled = False	(BoolOpt) Support an Alteon interswitch link for stateful session failover
service_ssl_throughput = 100	(IntOpt) service ssl throughput
service_throughput = 1000	(IntOpt) service throughput
vdirect_address = None	(StrOpt) vdirect server IP address
vdirect_password = radware	(StrOpt) vdirect user password
vdirect_user = vDirect	(StrOpt) vdirect user name

Logging

Use the following options to alter logging settings.

Table 7.51. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
default_log_levels = amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN	(ListOpt) list of logger=LEVEL pairs
fatal_deprecations = False	(BoolOpt) make deprecations fatal
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s	(StrOpt) format string to use for log messages with context

Configuration option = Default value	Description
[% (request_id)s %(user_identity)s] %(instance)s %(message)s	
logging_debug_format_suffix = %(funcName)s %(pathname)s: %(lineno)d	(StrOpt) data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) prefix each line of exception output with this format
publish_errors = False	(BoolOpt) publish error events
syslog_log_facility = LOG_USER	(StrOpt) syslog facility to receive log lines
use_ssl = False	(BoolOpt) Enable SSL on the API server
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Metadata Agent

Use the following options in the `metadata_agent.ini` file for the Metadata agent.

Table 7.52. Description of configuration options for metadata

Configuration option = Default value	Description
[DEFAULT]	
meta_flavor_driver_mappings = None	(StrOpt) Mapping between flavor and LinuxInterfaceDriver
metadata_backlog = 128	(IntOpt) Number of backlog requests to configure the metadata server socket with
metadata_port = 9697	(IntOpt) TCP Port used by Neutron metadata namespace proxy.
metadata_proxy_shared_secret =	(StrOpt) Shared secret to sign instance-id request
metadata_proxy_socket = \$state_path/metadata_proxy	(StrOpt) Location of Metadata Proxy UNIX domain socket
metadata_workers = 0	(IntOpt) Number of separate worker processes for metadata server
nova_metadata_ip = 127.0.0.1	(StrOpt) IP address used by Nova metadata server.
nova_metadata_port = 8775	(IntOpt) TCP Port used by Nova metadata server.

Metering Agent

Use the following options in the `metering_agent.ini` file for the Metering agent.

Table 7.53. Description of configuration options for metering_agent

Configuration option = Default value	Description
[DEFAULT]	
driver = neutron.services.metering.drivers.noop.noop_driver.NoopMeteringDriver	(StrOpt) Metering driver
measure_interval = 30	(IntOpt) Interval between two metering measures
[AGENT]	

Configuration option = Default value	Description
report_interval = 30	(FloatOpt) Seconds between nodes reporting state to server; should be less than agent_down_time, best if it is half or less than agent_down_time.

Policy

Use the following options in the `neutron.conf` file to change policy settings.

Table 7.54. Description of configuration options for policy

Configuration option = Default value	Description
[DEFAULT]	
allow_overlapping_ips = False	(BoolOpt) Allow overlapping IP support in Neutron
policy_file = policy.json	(StrOpt) The policy file to use

Quotas

Use the following options in the `neutron.conf` file for the quota system.

Table 7.55. Description of configuration options for quotas

Configuration option = Default value	Description
[DEFAULT]	
max_routes = 30	(IntOpt) Maximum number of routes
[QUOTAS]	
default_quota = -1	(IntOpt) Default number of resource allowed per tenant. A negative value means unlimited.
quota_driver = neutron.db.quota_db.DbQuotaDriver	(StrOpt) Default driver to use for quota checks
quota_firewall = 1	(IntOpt) Number of firewalls allowed per tenant. A negative value means unlimited.
quota_firewall_policy = 1	(IntOpt) Number of firewall policies allowed per tenant. A negative value means unlimited.
quota_firewall_rule = -1	(IntOpt) Number of firewall rules allowed per tenant. A negative value means unlimited.
quota_floatingip = 50	(IntOpt) Number of floating IPs allowed per tenant. A negative value means unlimited.
quota_health_monitor = -1	(IntOpt) Number of health monitors allowed per tenant. A negative value means unlimited.
quota_items = network, subnet, port	(ListOpt) Resource name(s) that are supported in quota features
quota_member = -1	(IntOpt) Number of pool members allowed per tenant. A negative value means unlimited.
quota_network = 10	(IntOpt) Number of networks allowed per tenant. A negative value means unlimited.
quota_network_gateway = 5	(IntOpt) Number of network gateways allowed per tenant, -1 for unlimited
quota_packet_filter = 100	(IntOpt) Number of packet_filters allowed per tenant, -1 for unlimited
quota_pool = 10	(IntOpt) Number of pools allowed per tenant. A negative value means unlimited.
quota_port = 50	(IntOpt) Number of ports allowed per tenant. A negative value means unlimited.

Configuration option = Default value	Description
quota_router = 10	(IntOpt) Number of routers allowed per tenant. A negative value means unlimited.
quota_security_group = 10	(IntOpt) Number of security groups allowed per tenant. A negative value means unlimited.
quota_security_group_rule = 100	(IntOpt) Number of security rules allowed per tenant. A negative value means unlimited.
quota_subnet = 10	(IntOpt) Number of subnets allowed per tenant, A negative value means unlimited.
quota_vip = 10	(IntOpt) Number of vips allowed per tenant. A negative value means unlimited.

Rootwrap

Use the following options in the `neutron.conf` file for the rootwrap settings

Table 7.56. Description of configuration options for rootwrap

Configuration option = Default value	Description
[DEFAULT]	
filters_path = /etc/neutron/rootwrap.d,/usr/share/neutron/rootwrap,/etc/quantum/rootwrap.d,/usr/share/quantum/rootwrap	List of directories to load filter definitions from (separated by ','). These directories MUST all be only writeable by root !
exec_dirs = /sbin,/usr/sbin,/bin,/usr/bin	List of directories to search executables in, in case filters do not explicitly specify a full path (separated by ',') If not specified, defaults to system PATH environment variable. These directories MUST all be only writeable by root !
use_syslog = False	Enable logging to syslog Default value is False
syslog_log_facility = syslog	Which syslog facility to use. Valid values include auth, authpriv, syslog, local0, local1... Default value is 'syslog'
syslog_log_level = ERROR	Which messages to log. INFO means log all usage ERROR means only log unsuccessful attempts
[xenapi]	
xenapi_connection_url = <None>	XenAPI configuration is only required by the L2 agent if it is to target a XenServer/XCP compute host's dom0.
xenapi_connection_username = root	No help text available for this option.
xenapi_connection_password = <None>	No help text available for this option.

Scheduler

Use the following options in the `neutron.conf` file to change scheduler settings.

Table 7.57. Description of configuration options for scheduler

Configuration option = Default value	Description
[DEFAULT]	
network_auto_schedule = True	(BoolOpt) Allow auto scheduling networks to DHCP agent.
network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.ChanceScheduler	(StrOpt) Driver to use for scheduling network to DHCP agent
router_auto_schedule = True	(BoolOpt) Allow auto scheduling of routers to L3 agent.
router_delete_namespaces = False	(BoolOpt) Delete namespace after removing a router.
router_scheduler_driver = neutron.scheduler.l3_agent_scheduler.ChanceScheduler	(StrOpt) Driver to use for scheduling router to a default L3 agent

Security Groups

Use the following options in the configuration file for your driver to change security group settings.

Table 7.58. Description of configuration options for securitygroups

Configuration option = Default value	Description
[SECURITYGROUP]	
enable_security_group = True	(BoolOpt) Controls whether the neutron security group API is enabled in the server. It should be false when using no security groups or using the nova security group API.
firewall_driver = None	(StrOpt) Driver for security groups firewall in the L2 agent

SSL

Use the following options in the `neutron.conf` file to enable SSL.

Table 7.59. Description of configuration options for ssl

Configuration option = Default value	Description
[DEFAULT]	
ssl_ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
ssl_cert_file = None	(StrOpt) Certificate file to use when starting the server securely
ssl_key_file = None	(StrOpt) Private key file to use when starting the server securely
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Testing

Use the following options to alter testing-related features.

Table 7.60. Description of configuration options for testing

Configuration option = Default value	Description
[DEFAULT]	
backdoor_port = None	(StrOpt) 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 service's log file.
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider

vArmour Firewall-as-a-Service driver

Use the following options in the `l3_agent.ini` file for the vArmour FwaaS driver.

Table 7.61. Description of configuration options for varmour

Configuration option = Default value	Description
[vArmour]	
director = localhost	(StrOpt) vArmour director ip
director_port = 443	(StrOpt) vArmour director port
password = varmour	(StrOpt) vArmour director password
username = varmour	(StrOpt) vArmour director username

VPN

Use the following options in the `vpn_agent.ini` file for the VPN agent.

Table 7.62. Description of configuration options for vpn

Configuration option = Default value	Description
[ipsec]	
config_base_dir = \$state_path/ipsec	(StrOpt) Location to store ipsec server config files
ipsec_status_check_interval = 60	(IntOpt) Interval for checking ipsec status
[openswan]	
ipsec_config_template = /usr/lib/python/site-packages/neutron/services/vpn/device_drivers/template/openswan/ipsec.conf.template	(StrOpt) Template file for ipsec configuration
ipsec_secret_template = /usr/lib/python/site-packages/neutron/services/vpn/device_drivers/template/openswan/ipsec.secret.template	(StrOpt) Template file for ipsec secret configuration
[vpnagent]	
vpn_device_driver = ['neutron.services.vpn.device_drivers.ipsec.OpenSwanDriver']	(MultiStrOpt) The vpn device drivers Neutron will use

WSGI

Use the following options in the `neutron.conf` file to configure the WSGI layer.

Table 7.63. Description of configuration options for wsgi

Configuration option = Default value	Description
[DEFAULT]	
backlog = 4096	(IntOpt) Number of backlog requests to configure the socket with
retry_until_window = 30	(IntOpt) Number of seconds to keep retrying to listen
tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.

Log files used by Networking

The corresponding log file of each Networking service is stored in the `/var/log/neutron/` directory of the host on which each service runs.

Table 7.64. Log files used by Networking services

Log file	Service/interface
dhcp-agent.log	neutron-dhcp-agent
l3-agent.log	neutron-l3-agent
lbaas-agent.log	neutron-lbaas-agent ^a
linuxbridge-agent.log	neutron-linuxbridge-agent
metadata-agent.log	neutron-metadata-agent
metering-agent.log	neutron-metering-agent
openvswitch-agent.log	neutron-openvswitch-agent
server.log	neutron-server

^aThe neutron-lbaas-agent service only runs when Load Balancer as a Service is enabled.

Networking sample configuration files

All the files in this section can be found in `/etc/neutron/`.

neutron.conf

Use the `neutron.conf` file to configure the majority of the OpenStack Networking options.

```
[DEFAULT]
# Print more verbose output (set logging level to INFO instead of default
# WARNING level).
# verbose = False

# Print debugging output (set logging level to DEBUG instead of default
# WARNING level).
# debug = False

# Where to store Neutron state files. This directory must be writable by the
# user executing the agent.
# state_path = /var/lib/neutron

# Where to store lock files
lock_path = $state_path/lock

# log_format = %(asctime)s %(levelname)8s [%(name)s] %(message)s
# log_date_format = %Y-%m-%d %H:%M:%S

# use_syslog -> syslog
# log_file and log_dir -> log_dir/log_file
# (not log_file) and log_dir -> log_dir/{binary_name}.log
# use_stderr -> stderr
# (not user_stderr) and (not log_file) -> stdout
# publish_errors -> notification system

# use_syslog = False
# syslog_log_facility = LOG_USER

# use_stderr = True
# log_file =
# log_dir =
```

```
# publish_errors = False

# Address to bind the API server to
# bind_host = 0.0.0.0

# Port to bind the API server to
# bind_port = 9696

# Path to the extensions. Note that this can be a colon-separated list of
# paths. For example:
# api_extensions_path = extensions:/path/to/more/extensions:/even/more/
# extensions
# The __path__ of neutron.extensions is appended to this, so if your
# extensions are in there you don't need to specify them here
# api_extensions_path =

# (StrOpt) Neutron core plugin entrypoint to be loaded from the
# neutron.core_plugins namespace. See setup.cfg for the entrypoint names of
# the
# plugins included in the neutron source distribution. For compatibility with
# previous versions, the class name of a plugin can be specified instead of
# its
# entrypoint name.
#
# core_plugin =
# Example: core_plugin = ml2

# (ListOpt) List of service plugin entrypoints to be loaded from the
# neutron.service_plugins namespace. See setup.cfg for the entrypoint names of
# the plugins included in the neutron source distribution. For compatibility
# with previous versions, the class name of a plugin can be specified instead
# of its entrypoint name.
#
# service_plugins =
# Example: service_plugins = router,firewall,lbaas,vpnaas,metering

# Paste configuration file
# api_paste_config = api-paste.ini

# The strategy to be used for auth.
# Supported values are 'keystone'(default), 'noauth'.
# auth_strategy = keystone

# Base MAC address. The first 3 octets will remain unchanged. If the
# 4th octet is not 00, it will also be used. The others will be
# randomly generated.
# 3 octet
# base_mac = fa:16:3e:00:00:00
# 4 octet
# base_mac = fa:16:3e:4f:00:00

# Maximum amount of retries to generate a unique MAC address
# mac_generation_retries = 16

# DHCP Lease duration (in seconds)
# dhcp_lease_duration = 86400

# Allow sending resource operation notification to DHCP agent
# dhcp_agent_notification = True
```

```
# Enable or disable bulk create/update/delete operations
# allow_bulk = True
# Enable or disable pagination
# allow_pagination = False
# Enable or disable sorting
# allow_sorting = False
# Enable or disable overlapping IPs for subnets
# Attention: the following parameter MUST be set to False if Neutron is
# being used in conjunction with nova security groups
# allow_overlapping_ips = False
# Ensure that configured gateway is on subnet
# force_gateway_on_subnet = False

# RPC configuration options. Defined in rpc __init__
# The messaging module to use, defaults to kombu.
# rpc_backend = neutron.openstack.common.rpc.impl_kombu
# Size of RPC thread pool
# rpc_thread_pool_size = 64
# Size of RPC connection pool
# rpc_conn_pool_size = 30
# Seconds to wait for a response from call or multicall
# rpc_response_timeout = 60
# Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
# rpc_cast_timeout = 30
# Modules of exceptions that are permitted to be recreated
# upon receiving exception data from an rpc call.
# allowed_rpc_exception_modules = neutron.openstack.common.exception, nova.
exception
# AMQP exchange to connect to if using RabbitMQ or QPID
# control_exchange = neutron

# If passed, use a fake RabbitMQ provider
# fake_rabbit = False

# Configuration options if sending notifications via kombu rpc (these are
# the defaults)
# SSL version to use (valid only if SSL enabled)
# kombu_ssl_version =
# SSL key file (valid only if SSL enabled)
# kombu_ssl_keyfile =
# SSL cert file (valid only if SSL enabled)
# kombu_ssl_certfile =
# SSL certification authority file (valid only if SSL enabled)
# kombu_ssl_ca_certs =
# IP address of the RabbitMQ installation
# rabbit_host = localhost
# Password of the RabbitMQ server
# rabbit_password = guest
# Port where RabbitMQ server is running/listening
# rabbit_port = 5672
# RabbitMQ single or HA cluster (host:port pairs i.e: host1:5672, host2:5672)
# rabbit_hosts is defaulted to '$rabbit_host:$rabbit_port'
# rabbit_hosts = localhost:5672
# User ID used for RabbitMQ connections
# rabbit_userid = guest
# Location of a virtual RabbitMQ installation.
# rabbit_virtual_host = /
# Maximum retries with trying to connect to RabbitMQ
# (the default of 0 implies an infinite retry count)
```

```
# rabbit_max_retries = 0
# RabbitMQ connection retry interval
# rabbit_retry_interval = 1
# Use HA queues in RabbitMQ (x-ha-policy: all). You need to
# wipe RabbitMQ database when changing this option. (boolean value)
# rabbit_ha_queues = false

# QPID
# rpc_backend=neutron.openstack.common.rpc.impl_qpid
# Qpid broker hostname
# qpid_hostname = localhost
# Qpid broker port
# qpid_port = 5672
# Qpid single or HA cluster (host:port pairs i.e: host1:5672, host2:5672)
# qpid_hosts is defaulted to '$qpid_hostname:$qpid_port'
# qpid_hosts = localhost:5672
# Username for qpid connection
# qpid_username = ''
# Password for qpid connection
# qpid_password = ''
# Space separated list of SASL mechanisms to use for auth
# qpid_sasl_mechanisms = ''
# Seconds between connection keepalive heartbeats
# qpid_heartbeat = 60
# Transport to use, either 'tcp' or 'ssl'
# qpid_protocol = tcp
# Disable Nagle algorithm
# qpid_tcp_nodelay = True

# ZMQ
# rpc_backend=neutron.openstack.common.rpc.impl_zmq
# ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP.
# The "host" option should point or resolve to this address.
# rpc_zmq_bind_address = *

# ===== Notification System Options =====

# Notifications can be sent when network/subnet/port are created, updated or
# deleted.
# There are three methods of sending notifications: logging (via the
# log_file directive), rpc (via a message queue) and
# noop (no notifications sent, the default)

# Notification_driver can be defined multiple times
# Do nothing driver
# notification_driver = neutron.openstack.common.notifier.no_op_notifier
# Logging driver
# notification_driver = neutron.openstack.common.notifier.log_notifier
# RPC driver.
notification_driver = neutron.openstack.common.notifier.rpc_notifier

# default_notification_level is used to form actual topic name(s) or to set
# logging level
# default_notification_level = INFO

# default_publisher_id is a part of the notification payload
# host = myhost.com
# default_publisher_id = $host

# Defined in rpc_notifier, can be comma separated values.
```



```
# worker thread in the current process. Greater than 0 launches that number
of
# child processes as RPC workers. The parent process manages them.
# This feature is experimental until issues are addressed and testing has been
# enabled for various plugins for compatibility.
# rpc_workers = 0

# wsgi keepalive option. Determines if connections are allowed to be held open
# by clients after a request is fulfilled. A value of False will ensure that
# the socket connection will be explicitly closed once a response has been
# sent to the client.
# wsgi_keep_alive = True

# Sets the value of TCP_KEEPIDLE in seconds to use for each server socket when
# starting API server. Not supported on OS X.
# tcp_keepidle = 600

# Number of seconds to keep retrying to listen
# retry_until_window = 30

# Number of backlog requests to configure the socket with.
# backlog = 4096

# Max header line to accommodate large tokens
# max_header_line = 16384

# Enable SSL on the API server
# use_ssl = False

# Certificate file to use when starting API server securely
# ssl_cert_file = /path/to/certfile

# Private key file to use when starting API server securely
# ssl_key_file = /path/to/keyfile

# CA certificate file to use when starting API server securely to
# verify connecting clients. This is an optional parameter only required if
# API clients need to authenticate to the API server using SSL certificates
# signed by a trusted CA
# ssl_ca_file = /path/to/cafile
# ===== end of WSGI parameters related to the API server =====

# ===== neutron nova interactions =====
# Send notification to nova when port status is active.
# notify_nova_on_port_status_changes = True

# Send notifications to nova when port data (fixed_ips/floatingips) change
# so nova can update it's cache.
# notify_nova_on_port_data_changes = True

# URL for connection to nova (Only supports one nova region currently).
# nova_url = http://127.0.0.1:8774/v2

# Name of nova region to use. Useful if keystone manages more than one region
# nova_region_name =

# Username for connection to nova in admin context
# nova_admin_username =
```

```
# The uuid of the admin nova tenant
# nova_admin_tenant_id =

# Password for connection to nova in admin context.
# nova_admin_password =

# Authorization URL for connection to nova in admin context.
# nova_admin_auth_url =

# Number of seconds between sending events to nova if there are any events to
send
# send_events_interval = 2

# ===== end of neutron nova interactions =====

[quotas]
# Default driver to use for quota checks
# quota_driver = neutron.db.quota_db.DbQuotaDriver

# Resource name(s) that are supported in quota features
# quota_items = network,subnet,port

# Default number of resource allowed per tenant. A negative value means
# unlimited.
# default_quota = -1

# Number of networks allowed per tenant. A negative value means unlimited.
# quota_network = 10

# Number of subnets allowed per tenant. A negative value means unlimited.
# quota_subnet = 10

# Number of ports allowed per tenant. A negative value means unlimited.
# quota_port = 50

# Number of security groups allowed per tenant. A negative value means
# unlimited.
# quota_security_group = 10

# Number of security group rules allowed per tenant. A negative value means
# unlimited.
# quota_security_group_rule = 100

# Number of vips allowed per tenant. A negative value means unlimited.
# quota_vip = 10

# Number of pools allowed per tenant. A negative value means unlimited.
# quota_pool = 10

# Number of pool members allowed per tenant. A negative value means unlimited.
# The default is unlimited because a member is not a real resource consumer
# on Openstack. However, on back-end, a member is a resource consumer
# and that is the reason why quota is possible.
# quota_member = -1

# Number of health monitors allowed per tenant. A negative value means
# unlimited.
# The default is unlimited because a health monitor is not a real resource
# consumer on Openstack. However, on back-end, a member is a resource consumer
# and that is the reason why quota is possible.
```



```
# Verbosity of SQL debugging information. 0=None, 100=Everything
# connection_debug = 0

# Add python stack traces to SQL as comment strings
# connection_trace = False

# If set, use this value for pool_timeout with sqlalchemy
# pool_timeout = 10

[service_providers]
# Specify service providers (drivers) for advanced services like loadbalancer,
# VPN, Firewall.
# Must be in form:
# service_provider=<service_type>:<name>:<driver>[:default]
# List of allowed service types includes LOADBALANCER, FIREWALL, VPN
# Combination of <service type> and <name> must be unique; <driver> must also
# be unique
# This is multiline option, example for default provider:
# service_provider=LOADBALANCER:name:lbaas_plugin_driver_path:default
# example of non-default provider:
# service_provider=FIREWALL:name2:firewall_driver_path
# --- Reference implementations ---
service_provider=LOADBALANCER:Haproxy:neutron.services.loadbalancer.drivers.
haproxy.plugin_driver.HaproxyOnHostPluginDriver:default
service_provider=VPN:openswan:neutron.services.vpn.service_drivers.ipsec.
IPsecVPNDriver:default
# In order to activate Radware's lbaas driver you need to uncomment the next
# line.
# If you want to keep the HA Proxy as the default lbaas driver, remove the
# attribute default from the line below.
# Otherwise comment the HA Proxy line
# service_provider = LOADBALANCER:Radware:neutron.services.loadbalancer.
drivers.radware.driver.LoadBalancerDriver:default
# uncomment the following line to make the 'netscaler' LBaaS provider
# available.
# service_provider=LOADBALANCER:NetScaler:neutron.services.loadbalancer.
drivers.netscaler.netscaler_driver.NetScalerPluginDriver
# Uncomment the following line (and comment out the OpenSwan VPN line) to
# enable Cisco's VPN driver.
# service_provider=VPN:cisco:neutron.services.vpn.service_drivers.cisco_ipsec.
CiscoCsrIPsecVPNDriver:default
# Uncomment the line below to use Embrane heleos as Load Balancer service
# provider.
# service_provider=LOADBALANCER:Embrane:neutron.services.loadbalancer.drivers.
embrane.driver.EmbraneLbaas:default
```

api-paste.ini

Use the `api-paste.ini` to configure the OpenStack Networking API.

```
[composite:neutron]
use = egg:Paste#urlmap
/: neutronversions
/v2.0: neutronapi_v2_0

[composite:neutronapi_v2_0]
```

```
use = call:neutron.auth:pipeline_factory
noauth = request_id catch_errors extensions neutronapiapp_v2_0
keystone = request_id catch_errors authtoken keystonecontext extensions
           neutronapiapp_v2_0

[filter:request_id]
paste.filter_factory = neutron.openstack.common.middleware.
request_id:RequestIdMiddleware.factory

[filter:catch_errors]
paste.filter_factory = neutron.openstack.common.middleware.
catch_errors:CatchErrorsMiddleware.factory

[filter:keystonecontext]
paste.filter_factory = neutron.auth:NeutronKeystoneContext.factory

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory

[filter:extensions]
paste.filter_factory = neutron.api.
extensions:plugin_aware_extension_middleware_factory

[app:neutronversions]
paste.app_factory = neutron.api.versions:Versions.factory

[app:neutronapiapp_v2_0]
paste.app_factory = neutron.api.v2.router:APIRouter.factory
```

policy.json

Use the `policy.json` file to define additional access controls that apply to the OpenStack Networking service.

```
{
  "context_is_admin": "role:admin",
  "admin_or_owner": "rule:context_is_admin or tenant_id:$(tenant_id)s",
  "admin_or_network_owner": "rule:context_is_admin or tenant_id:
$(network:tenant_id)s",
  "admin_only": "rule:context_is_admin",
  "regular_user": "",
  "shared": "field:networks:shared=True",
  "shared_firewalls": "field:firewalls:shared=True",
  "external": "field:networks:router:external=True",
  "default": "rule:admin_or_owner",

  "subnets:private:read": "rule:admin_or_owner",
  "subnets:private:write": "rule:admin_or_owner",
  "subnets:shared:read": "rule:regular_user",
  "subnets:shared:write": "rule:admin_only",

  "create_subnet": "rule:admin_or_network_owner",
  "get_subnet": "rule:admin_or_owner or rule:shared",
  "update_subnet": "rule:admin_or_network_owner",
  "delete_subnet": "rule:admin_or_network_owner",
```

```
"create_network": "",
"get_network": "rule:admin_or_owner or rule:shared or rule:external",
"get_network:router:external": "rule:regular_user",
"get_network:segments": "rule:admin_only",
"get_network:provider:network_type": "rule:admin_only",
"get_network:provider:physical_network": "rule:admin_only",
"get_network:provider:segmentation_id": "rule:admin_only",
"get_network:queue_id": "rule:admin_only",
"create_network:shared": "rule:admin_only",
"create_network:router:external": "rule:admin_only",
"create_network:segments": "rule:admin_only",
"create_network:provider:network_type": "rule:admin_only",
"create_network:provider:physical_network": "rule:admin_only",
"create_network:provider:segmentation_id": "rule:admin_only",
"update_network": "rule:admin_or_owner",
"update_network:segments": "rule:admin_only",
"update_network:shared": "rule:admin_only",
"update_network:provider:network_type": "rule:admin_only",
"update_network:provider:physical_network": "rule:admin_only",
"update_network:provider:segmentation_id": "rule:admin_only",
"delete_network": "rule:admin_or_owner",

"create_port": "",
"create_port:mac_address": "rule:admin_or_network_owner",
"create_port:fixed_ips": "rule:admin_or_network_owner",
"create_port:port_security_enabled": "rule:admin_or_network_owner",
"create_port:binding:host_id": "rule:admin_only",
"create_port:binding:profile": "rule:admin_only",
"create_port:mac_learning_enabled": "rule:admin_or_network_owner",
"get_port": "rule:admin_or_owner",
"get_port:queue_id": "rule:admin_only",
"get_port:binding:vif_type": "rule:admin_only",
"get_port:binding:vif_details": "rule:admin_only",
"get_port:binding:host_id": "rule:admin_only",
"get_port:binding:profile": "rule:admin_only",
"update_port": "rule:admin_or_owner",
"update_port:fixed_ips": "rule:admin_or_network_owner",
"update_port:port_security_enabled": "rule:admin_or_network_owner",
"update_port:binding:host_id": "rule:admin_only",
"update_port:binding:profile": "rule:admin_only",
"update_port:mac_learning_enabled": "rule:admin_or_network_owner",
"delete_port": "rule:admin_or_owner",

"create_router:external_gateway_info:enable_snat": "rule:admin_only",
"update_router:external_gateway_info:enable_snat": "rule:admin_only",

"create_firewall": "",
"get_firewall": "rule:admin_or_owner",
"create_firewall:shared": "rule:admin_only",
"get_firewall:shared": "rule:admin_only",
"update_firewall": "rule:admin_or_owner",
"delete_firewall": "rule:admin_or_owner",

"create_firewall_policy": "",
"get_firewall_policy": "rule:admin_or_owner or rule:shared_firewalls",
"create_firewall_policy:shared": "rule:admin_or_owner",
"update_firewall_policy": "rule:admin_or_owner",
"delete_firewall_policy": "rule:admin_or_owner",

"create_firewall_rule": "",
```


rootwrap.conf

Use the `rootwrap.conf` file to define configuration values used by the `rootwrap` script when the OpenStack Networking service must escalate its privileges to those of the root user.

```
# Configuration for neutron-rootwrap
# This file should be owned by (and only-writeable by) the root user

[DEFAULT]
# List of directories to load filter definitions from (separated by ',').
# These directories MUST all be only writeable by root !
filters_path=/etc/neutron/rootwrap.d,/usr/share/neutron/rootwrap,/etc/quantum/
rootwrap.d,/usr/share/quantum/rootwrap

# List of directories to search executables in, in case filters do not
# explicitly specify a full path (separated by ',')
# If not specified, defaults to system PATH environment variable.
# These directories MUST all be only writeable by root !
exec_dirs=/sbin,/usr/sbin,/bin,/usr/bin

# Enable logging to syslog
# Default value is False
use_syslog=False

# Which syslog facility to use.
# Valid values include auth, authpriv, syslog, local0, local1...
# Default value is 'syslog'
syslog_log_facility=syslog

# Which messages to log.
# INFO means log all usage
# ERROR means only log unsuccessful attempts
syslog_log_level=ERROR

[xenapi]
# XenAPI configuration is only required by the L2 agent if it is to
# target a XenServer/XCP compute host's dom0.
xenapi_connection_url=<None>
xenapi_connection_username=root
xenapi_connection_password=<None>
```

Configuration files for plug-in agents

Each plug-in agent that runs on an OpenStack Networking node, to perform local networking configuration for the node's VMs and networking services, has its own configuration file.

dhcp_agent.ini

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
# debug = False
```



```
# The DHCP agent will resync its state with Neutron to recover from any
# transient notification or rpc errors. The interval is number of
# seconds between attempts.
# resync_interval = 5

# The DHCP agent requires an interface driver be set. Choose the one that best
# matches your plugin.
# interface_driver =

# Example of interface_driver option for OVS based plugins(OVS, Ryu, NEC, NVP,
# BigSwitch/Floodlight)
# interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver

# Name of Open vSwitch bridge to use
# ovs_integration_bridge = br-int

# Use veth for an OVS interface or not.
# Support kernels with limited namespace support
# (e.g. RHEL 6.5) so long as ovs_use_veth is set to True.
# ovs_use_veth = False

# Example of interface_driver option for LinuxBridge
# interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver

# The agent can use other DHCP drivers. Dnsmasq is the simplest and requires
# no additional setup of the DHCP server.
# dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq

# Allow overlapping IP (Must have kernel build with CONFIG_NET_NS=y and
# iproute2 package that supports namespaces).
# use_namespaces = True

# The DHCP server can assist with providing metadata support on isolated
# networks. Setting this value to True will cause the DHCP server to append
# specific host routes to the DHCP request. The metadata service will only
# be activated when the subnet does not contain any router port. The guest
# instance must be configured to request host routes via DHCP (Option 121).
# enable_isolated_metadata = False

# Allows for serving metadata requests coming from a dedicated metadata
# access network whose cidr is 169.254.169.254/16 (or larger prefix), and
# is connected to a Neutron router from which the VMs send metadata
# request. In this case DHCP Option 121 will not be injected in VMs, as
# they will be able to reach 169.254.169.254 through a router.
# This option requires enable_isolated_metadata = True
# enable_metadata_network = False

# Number of threads to use during sync process. Should not exceed connection
# pool size configured on server.
# num_sync_threads = 4

# Location to store DHCP server config files
# dhcp_confs = $state_path/dhcp

# Domain to use for building the hostnames
# dhcp_domain = openstacklocal

# Override the default dnsmasq settings with this file
# dnsmasq_config_file =
```

```
# Comma-separated list of DNS servers which will be used by dnsmasq
# as forwarders.
# dnsmasq_dns_servers =

# Limit number of leases to prevent a denial-of-service.
# dnsmasq_lease_max = 16777216

# Location to DHCP lease relay UNIX domain socket
# dhcp_lease_relay_socket = $state_path/dhcp/lease_relay

# Location of Metadata Proxy UNIX domain socket
# metadata_proxy_socket = $state_path/metadata_proxy

# dhcp_delete_namespaces, which is false by default, can be set to True if
# namespaces can be deleted cleanly on the host running the dhcp agent.
# Do not enable this until you understand the problem with the Linux iproute
# utility mentioned in https://bugs.launchpad.net/neutron/+bug/1052535 and
# you are sure that your version of iproute does not suffer from the problem.
# If True, namespaces will be deleted when a dhcp server is disabled.
# dhcp_delete_namespaces = False

# Timeout for ovs-vsctl commands.
# If the timeout expires, ovs commands will fail with ALARMCLOCK error.
# ovs_vsctl_timeout = 10
```

l3_agent.ini

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
# debug = False

# L3 requires that an interface driver be set. Choose the one that best
# matches your plugin.
# interface_driver =

# Example of interface_driver option for OVS based plugins (OVS, Ryu, NEC)
# that supports L3 agent
# interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver

# Use veth for an OVS interface or not.
# Support kernels with limited namespace support
# (e.g. RHEL 6.5) so long as ovs_use_veth is set to True.
# ovs_use_veth = False

# Example of interface_driver option for LinuxBridge
# interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver

# Allow overlapping IP (Must have kernel build with CONFIG_NET_NS=y and
# iproute2 package that supports namespaces).
# use_namespaces = True

# If use_namespaces is set as False then the agent can only configure one
router.

# This is done by setting the specific router_id.
# router_id =
```

```
# When external_network_bridge is set, each L3 agent can be associated
# with no more than one external network. This value should be set to the UUID
# of that external network. To allow L3 agent support multiple external
# networks, both the external_network_bridge and gateway_external_network_id
# must be left empty.
# gateway_external_network_id =

# Indicates that this L3 agent should also handle routers that do not have
# an external network gateway configured. This option should be True only
# for a single agent in a Neutron deployment, and may be False for all agents
# if all routers must have an external network gateway
# handle_internal_only_routers = True

# Name of bridge used for external network traffic. This should be set to
# empty value for the linux bridge. when this parameter is set, each L3 agent
# can be associated with no more than one external network.
# external_network_bridge = br-ex

# TCP Port used by Neutron metadata server
# metadata_port = 9697

# Send this many gratuitous ARPs for HA setup. Set it below or equal to 0
# to disable this feature.
# send_arp_for_ha = 0

# seconds between re-sync routers' data if needed
# periodic_interval = 40

# seconds to start to sync routers' data after
# starting agent
# periodic_fuzzy_delay = 5

# enable_metadata_proxy, which is true by default, can be set to False
# if the Nova metadata server is not available
# enable_metadata_proxy = True

# Location of Metadata Proxy UNIX domain socket
# metadata_proxy_socket = $state_path/metadata_proxy

# router_delete_namespaces, which is false by default, can be set to True if
# namespaces can be deleted cleanly on the host running the L3 agent.
# Do not enable this until you understand the problem with the Linux iproute
# utility mentioned in https://bugs.launchpad.net/neutron/+bug/1052535 and
# you are sure that your version of iproute does not suffer from the problem.
# If True, namespaces will be deleted when a router is destroyed.
# router_delete_namespaces = False

# Timeout for ovs-vsctl commands.
# If the timeout expires, ovs commands will fail with ALARMCLOCK error.
# ovs_vsctl_timeout = 10
```

lbaas_agent.ini

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output).
# debug = False
```

```
# The LBaaS agent will resync its state with Neutron to recover from any
# transient notification or rpc errors. The interval is number of
# seconds between attempts.
# periodic_interval = 10

# LBaaS requires an interface driver be set. Choose the one that best
# matches your plugin.
# interface_driver =

# Example of interface_driver option for OVS based plugins (OVS, Ryu, NEC,
# NVP,
# BigSwitch/Floodlight)
# interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver

# Use veth for an OVS interface or not.
# Support kernels with limited namespace support
# (e.g. RHEL 6.5) so long as ovs_use_veth is set to True.
# ovs_use_veth = False

# Example of interface_driver option for LinuxBridge
# interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver

# The agent requires drivers to manage the loadbalancer. HAProxy is the
# opensource version.
# Multiple device drivers reflecting different service providers could be
# specified:
# device_driver = path.to.provider1.driver.Driver
# device_driver = path.to.provider2.driver.Driver
# Default is:
# device_driver = neutron.services.loadbalancer.drivers.haproxy.
namespace_driver.HaproxyNSDriver

[haproxy]
# Location to store config and state files
# loadbalancer_state_path = $state_path/lbaas

# The user group
# user_group = nogroup
```

metadata_agent.ini

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
# debug = True

# The Neutron user information for accessing the Neutron API.
auth_url = http://localhost:5000/v2.0
auth_region = RegionOne
# Turn off verification of the certificate for ssl
# auth_insecure = False
# Certificate Authority public key (CA cert) file for ssl
# auth_ca_cert =
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%

# Network service endpoint type to pull from the keystone catalog
```

```

# endpoint_type = adminURL

# IP address used by Nova metadata server
# nova_metadata_ip = 127.0.0.1

# TCP Port used by Nova metadata server
# nova_metadata_port = 8775

# When proxying metadata requests, Neutron signs the Instance-ID header with a
# shared secret to prevent spoofing. You may select any string for a secret,
# but it must match here and in the configuration used by the Nova Metadata
# Server. NOTE: Nova uses a different key:
# neutron_metadata_proxy_shared_secret
# metadata_proxy_shared_secret =

# Location of Metadata Proxy UNIX domain socket
# metadata_proxy_socket = $state_path/metadata_proxy

# Number of separate worker processes for metadata server
# metadata_workers = 0

# Number of backlog requests to configure the metadata server socket with
# metadata_backlog = 128

# URL to connect to the cache backend.
# Example of URL using memory caching backend
# with ttl set to 5 seconds: cache_url = memory://?default_ttl=5
# default_ttl=0 parameter will cause cache entries to never expire.
# Otherwise default_ttl specifies time in seconds a cache entry is valid for.
# No cache is used in case no value is passed.
# cache_url =

```

New, updated and deprecated options for neutron

Table 7.65. New options

Option = default value	(Type) Help string
auth_ca_cert = None	(StrOpt) Certificate Authority public key (CA cert) file for ssl
auth_insecure = False	(BoolOpt) Turn off verification of the certificate for ssl
ca_certs = None	(StrOpt) CA certificates
ctl_cert = None	(StrOpt) controller certificate
ctl_privkey = None	(StrOpt) controller private key
default_service_cluster_uuid = None	(StrOpt) Unique identifier of the Service Cluster which will be used by logical services like dhcp and metadata
dnsmasq_dns_servers = None	(ListOpt) Comma-separated list of the DNS servers which will be used as forwarders.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.

Option = default value	(Type) Help string
max_header_line = 16384	(IntOpt) Max header line to accommodate large tokens
max_request_body_size = 114688	(IntOpt) the maximum body size per each request(bytes)
metadata_backlog = 128	(IntOpt) Number of backlog requests to configure the metadata server socket with
metadata_workers = 0	(IntOpt) Number of separate worker processes for metadata server
notify_nova_on_port_data_changes = True	(BoolOpt) Send notification to nova when port data (fixed_ips/floatingip) changes so nova can update its cache.
notify_nova_on_port_status_changes = True	(BoolOpt) Send notification to nova when port status changes
nova_admin_auth_url = http://localhost:5000/v2.0	(StrOpt) Authorization URL for connecting to nova in admin context
nova_admin_password = None	(StrOpt) Password for connection to nova in admin context
nova_admin_tenant_id = None	(StrOpt) The uuid of the admin nova tenant
nova_admin_username = None	(StrOpt) Username for connecting to nova in admin context
nova_region_name = None	(StrOpt) Name of nova region to use. Useful if keystone manages more than one region.
nova_url = http://127.0.0.1:8774/v2	(StrOpt) URL for connection to nova
nsx_controllers = None	(ListOpt) Lists the NSX controllers in this cluster
nsx_password = admin	(StrOpt) Password for NSX controllers in this cluster
nsx_user = admin	(StrOpt) User name for NSX controllers in this cluster
ofp_listen_host =	(StrOpt) openflow listen host
ofp_ssl_listen_port = 6633	(IntOpt) openflow ssl listen port
ofp_tcp_listen_port = 6633	(IntOpt) openflow tcp listen port
ovs_vsctl_timeout = 10	(IntOpt) Timeout in seconds for ovs-vsctl commands
rpc_workers = 0	(IntOpt) Number of RPC worker processes for service
send_events_interval = 2	(IntOpt) Number of seconds between sending events to nova if there are any events to send.
AGENT/get_datapath_retry_times = 60	(IntOpt) Number of seconds to retry acquiring an Open vSwitch datapath
AGENT/integration_bridge = br-int	(StrOpt) integration bridge
CISCO/nexus_l3_enable = False	(BoolOpt) Enable L3 support on the Nexus switches
ESWITCH/apply_profile_patch = False	(BoolOpt) Enable server compatibility with old nova
META/rpc_flavor =	(StrOpt) Flavor of which plugin handles RPC
MLNX/physical_network_type = eth	(StrOpt) Physical network type for provider network (eth or ib)
MLNX/physical_network_type_mappings = []	(ListOpt) List of <physical_network>:<physical_network_type> with physical_network_type is either eth or ib
NOVA/node_override_vif_hostdev = []	(ListOpt) Nova compute nodes to manually set VIF type to hostdev
NOVA/node_override_vif_midonet = []	(ListOpt) Nova compute nodes to manually set VIF type to midonet
NOVA/node_override_vif_mlnx_direct = []	(ListOpt) Nova compute nodes to manually set VIF type to mlnx_direct
NSX/agent_mode = agent	(StrOpt) The mode used to implement DHCP/metadata services.

Option = default value	(Type) Help string
NSX/concurrent_connections = 10	(IntOpt) Maximum concurrent connections to each NSX controller.
NSX/default_transport_type = stt	(StrOpt) The default network transport type to use (stt, gre, bridge, ipsec_gre, or ipsec_stt)
NSX/max_ip_per_bridged_ls = 5000	(IntOpt) Maximum number of ports of a logical switch on a bridged transport zone (default 5000)
NSX/max_ip_per_overlay_ls = 256	(IntOpt) Maximum number of ports of a logical switch on an overlay transport zone (default 256)
NSX/metadata_mode = access_network	(StrOpt) If set to access_network this enables a dedicated connection to the metadata proxy for metadata server access via Neutron router. If set to dhcp_host_route this enables host route injection via the dhcp agent. This option is only useful if running on a host that does not support namespaces otherwise access_network should be used.
NSX/nsx_gen_timeout = -1	(IntOpt) Number of seconds a generation id should be valid for (default -1 meaning do not time out)
NSX/replication_mode = service	(StrOpt) The default option leverages service nodes to perform packet replication though one could set to this to 'source' to perform replication locally. This is useful if one does not want to deploy a service node(s).
NSX_DHCP/default_lease_time = 43200	(IntOpt) Default DHCP lease time
NSX_DHCP/domain_name = openstacklocal	(StrOpt) Domain to use for building the hostnames
NSX_DHCP/extra_domain_name_servers = []	(ListOpt) Comma separated list of additional domain name servers
NSX_LSN/sync_on_missing_data = False	(BoolOpt) Pull LSN information from NSX in case it is missing from the local data store. This is useful to rebuild the local store in case of server recovery.
NSX_METADATA/metadata_server_address = 127.0.0.1	(StrOpt) IP address used by Metadata server.
NSX_METADATA/metadata_server_port = 8775	(IntOpt) TCP Port used by Metadata server.
NSX_METADATA/metadata_shared_secret =	(StrOpt) Shared secret to sign instance-id request
NSX_SYNC/always_read_status = False	(BoolOpt) Always read operational status from backend on show operations. Enabling this option might slow down the system.
NSX_SYNC/max_random_sync_delay = 0	(IntOpt) Maximum value for the additional random delay in seconds between runs of the state synchronization task
NSX_SYNC/min_chunk_size = 500	(IntOpt) Minimum number of resources to be retrieved from NSX during state synchronization
NSX_SYNC/min_sync_req_delay = 1	(IntOpt) Minimum delay, in seconds, between two state synchronization queries to NSX. It must not exceed state_sync_interval
NSX_SYNC/state_sync_interval = 10	(IntOpt) Interval in seconds between runs of the state synchronization task. Set it to 0 to disable it
OFC/api_max_attempts = 3	(IntOpt) Maximum attempts per OFC API request. NEC plugin retries API request to OFC when OFC returns ServiceUnavailable (503). The value must be greater than 0.
OFC/insecure_ssl = False	(BoolOpt) Disable SSL certificate verification
OFC/path_prefix =	(StrOpt) Base URL of OFC REST API. It is prepended to each API request.
QUOTAS/quota_health_monitor = -1	(IntOpt) Number of health monitors allowed per tenant. A negative value means unlimited.
QUOTAS/quota_member = -1	(IntOpt) Number of pool members allowed per tenant. A negative value means unlimited.

Option = default value	(Type) Help string
QUOTAS/quota_pool = 10	(IntOpt) Number of pools allowed per tenant. A negative value means unlimited.
QUOTAS/quota_vip = 10	(IntOpt) Number of vips allowed per tenant. A negative value means unlimited.
RESTPROXY/auth_resource =	(StrOpt) Nuage provided uri for initial authorization to access VSD
RESTPROXY/auto_sync_on_failure = True	(BoolOpt) If neutron fails to create a resource because the backend controller doesn't know of a dependency, the plugin automatically triggers a full data synchronization to the controller.
RESTPROXY/base_uri = /	(StrOpt) Nuage provided base uri to reach out to VSD
RESTPROXY/cache_connections = True	(BoolOpt) Re-use HTTP/HTTPS connections to the controller.
RESTPROXY/consistency_interval = 60	(IntOpt) Time between verifications that the backend controller database is consistent with Neutron
RESTPROXY/default_floatingip_quota = 254	(IntOpt) Per Net Partition quota of floating ips
RESTPROXY/default_net_partition_name = OpenStackDefaultNetPartition	(StrOpt) Default Network partition in which VSD will orchestrate network resources using openstack
RESTPROXY/no_ssl_validation = False	(BoolOpt) Disables SSL certificate validation for controllers
RESTPROXY/organization = system	(StrOpt) Organization name in which VSD will orchestrate network resources using openstack
RESTPROXY/server = localhost:8800	(StrOpt) IP Address and Port of Nuage's VSD server
RESTPROXY/serverauth = username:password	(StrOpt) Username and password for authentication
RESTPROXY/serverssl = False	(BoolOpt) Boolean for SSL connection with VSD server
RESTPROXY/ssl_cert_directory = /etc/neutron/plugins/bigswitch/ssl	(StrOpt) Directory containing ca_certs and host_certs certificate directories.
RESTPROXY/ssl_sticky = True	(BoolOpt) Trust and store the first certificate received for each controller address and use it to validate future connections to that address.
RESTPROXY/thread_pool_size = 4	(IntOpt) Maximum number of threads to spawn to handle large volumes of port creations.
RESTPROXYAGENT/integration_bridge = br-int	(StrOpt) Name of integration bridge on compute nodes used for security group insertion.
RESTPROXYAGENT/polling_interval = 5	(IntOpt) Seconds between agent checks for port changes
RESTPROXYAGENT/virtual_switch_type = ovs	(StrOpt) Virtual switch type.
SDNVE/base_url = /one/nb/v2/	(StrOpt) Base URL for SDN-VE controller REST API
SDNVE/controller_ips = ['127.0.0.1']	(ListOpt) List of IP addresses of SDN-VE controller(s)
SDNVE/default_tenant_type = OF	(StrOpt) Tenant type: OF (default) and OVERLAY
SDNVE/format = json	(StrOpt) SDN-VE request/response format
SDNVE/info = sdnve_info_string	(StrOpt) SDN-VE RPC subject
SDNVE/integration_bridge = None	(StrOpt) Integration bridge to use
SDNVE/interface_mappings = []	(ListOpt) List of <physical_network_name>:<interface_name>
SDNVE/of_signature = SDNVE-OF	(StrOpt) The string in tenant description that indicates the tenant is a OF tenant
SDNVE/out_of_band = True	(BoolOpt) Indicating if controller is out of band or not
SDNVE/overlay_signature = SDNVE-OVERLAY	(StrOpt) The string in tenant description that indicates the tenant is a OVERLAY tenant
SDNVE/password = admin	(StrOpt) SDN-VE administrator password
SDNVE/port = 8443	(StrOpt) SDN-VE controller port number

Option = default value	(Type) Help string
SDNVE/reset_bridge = True	(BoolOpt) Reset the integration bridge before use
SDNVE/use_fake_controller = False	(BoolOpt) If set to True uses a fake controller.
SDNVE/userid = admin	(StrOpt) SDN-VE administrator user id
SDNVE_AGENT/polling_interval = 2	(IntOpt) Agent polling interval if necessary
SDNVE_AGENT/root_helper = sudo	(StrOpt) Using root helper
SDNVE_AGENT/rpc = True	(BoolOpt) Whether using rpc
SECURITYGROUP/enable_security_group = True	(BoolOpt) Controls whether the neutron security group API is enabled in the server. It should be false when using no security groups or using the nova security group API.
audit/api_audit_map = api_audit_map.conf	(StrOpt) File containing mapping for api paths and service endpoints
audit/namespace = openstack	(StrOpt) namespace prefix for generated id
cisco_csr_ipsec/status_check_interval = 60	(IntOpt) Status check interval for Cisco CSR IPsec connections
database/db_inc_retry_interval = True	(BoolOpt) Whether to increase interval between db connection retries, up to db_max_retry_interval
database/db_max_retries = 20	(IntOpt) maximum db connection retries before error is raised. (setting -1 implies an infinite retry count)
database/db_max_retry_interval = 10	(IntOpt) max seconds between db connection retries, if db_inc_retry_interval is enabled
database/db_retry_interval = 1	(IntOpt) seconds between db connection retries
database/use_db_reconnect = False	(BoolOpt) Enable the experimental use of database reconnect on connection lost
haproxy/loadbalancer_state_path = \$state_path/lbaas	(StrOpt) Location to store config and state files
haproxy/user_group = nogroup	(StrOpt) The user group
heleoslb/admin_password = None	(StrOpt) ESM admin password.
heleoslb/admin_username = None	(StrOpt) ESM admin username.
heleoslb/async_requests = None	(BoolOpt) Define if the requests have run asynchronously or not
heleoslb/dummy_utif_id = None	(StrOpt) Dummy user traffic Security Zone id for LBs
heleoslb/esm_mgmt = None	(StrOpt) ESM management root address
heleoslb/inband_id = None	(StrOpt) In band Security Zone id for LBs
heleoslb/lb_flavor = small	(StrOpt) choose LB image flavor to use, accepted values: small, medium
heleoslb/lb_image = None	(StrOpt) Load Balancer image id (Embrane LB)
heleoslb/mgmt_id = None	(StrOpt) Management Security Zone id for LBs
heleoslb/oob_id = None	(StrOpt) Out of band Security Zone id for LBs
heleoslb/resource_pool_id = None	(StrOpt) Shared resource pool id
heleoslb/sync_interval = 60	(IntOpt) resource synchronization interval in seconds
ml2_brocade/address =	(StrOpt) The address of the host to SSH to
ml2_brocade/ostype = NOS	(StrOpt) Unused
ml2_brocade/password = password	(StrOpt) The SSH password to use
ml2_brocade/physical_networks =	(StrOpt) Allowed physical networks
ml2_brocade/username = admin	(StrOpt) The SSH username to use
ml2_odl/password = None	(StrOpt) HTTP password for authentication
ml2_odl/session_timeout = 30	(IntOpt) Tomcat session timeout in minutes.
ml2_odl/timeout = 10	(IntOpt) HTTP timeout in seconds.
ml2_odl/url = None	(StrOpt) HTTP URL of OpenDaylight REST interface.

Option = default value	(Type) Help string
ml2_odl/username = None	(StrOpt) HTTP username for authentication
netScaler_driver/netScaler_ncc_password = None	(StrOpt) Password to login to the NetScaler Control Center Server.
netScaler_driver/netScaler_ncc_uri = None	(StrOpt) The URL to reach the NetScaler Control Center Server.
netScaler_driver/netScaler_ncc_username = None	(StrOpt) Username to login to the NetScaler Control Center Server.
nvSd/nvSd_ip = 127.0.0.1	(StrOpt) NVSD Controller IP address
nvSd/nvSd_passwd = oc123	(StrOpt) NVSD Controller password
nvSd/nvSd_port = 8082	(IntOpt) NVSD Controller Port number
nvSd/nvSd_retries = 0	(IntOpt) Number of login retries to NVSD controller
nvSd/nvSd_user = ocplugin	(StrOpt) NVSD Controller username
nvSd/request_timeout = 30	(IntOpt) NVSD controller REST API request timeout in seconds
plumGriddirector/director_server = localhost	(StrOpt) PLUMgrid Director server to connect to
plumGriddirector/director_server_port = 8080	(StrOpt) PLUMgrid Director server port to connect to
plumGriddirector/password = password	(StrOpt) PLUMgrid Director admin password
plumGriddirector/servertimeout = 5	(IntOpt) PLUMgrid Director server timeout
plumGriddirector/username = username	(StrOpt) PLUMgrid Director admin username
radware/actions_to_skip = ['setup_I2_I3']	(ListOpt) List of actions that we dont want to push to the completion queue
radware/I2_I3_ctor_params = {'ha_ip_pool_name': 'default', 'allocate_ha_vrrp': True, 'ha_network_name': 'HA-Network', 'service': '_REPLACE_', 'allocate_ha_ips': True}	(DictOpt) I2_I3 workflow constructor params
radware/I2_I3_setup_params = {'data_ip_address': '192.168.200.99', 'data_port': 1, 'gateway': '192.168.200.1', 'ha_port': 2, 'data_ip_mask': '255.255.255.0'}	(DictOpt) I2_I3 workflow setup params
radware/I2_I3_workflow_name = openstack_I2_I3	(StrOpt) I2_I3 workflow name
radware/I4_action_name = BaseCreate	(StrOpt) I4 workflow action name
radware/I4_workflow_name = openstack_I4	(StrOpt) I4 workflow name
radware/service_adc_type = VA	(StrOpt) Service ADC type
radware/service_adc_version =	(StrOpt) Service ADC version
radware/service_cache = 20	(IntOpt) service cache
radware/service_compression_throughput = 100	(IntOpt) service compression throughput
radware/service_ha_pair = False	(BoolOpt) service HA pair
radware/service_isl_vlan = -1	(IntOpt) A required VLAN for the interswitch link to use
radware/service_resource_pool_ids = []	(ListOpt) Resource pool ids
radware/service_session_mirroring_enabled = False	(BoolOpt) Support an Alteon interswitch link for stateful session failover
radware/service_ssl_throughput = 100	(IntOpt) service ssl throughput
radware/service_throughput = 1000	(IntOpt) service throughput
radware/vdirect_address = None	(StrOpt) vdirect server IP address
radware/vdirect_password = radware	(StrOpt) vdirect user password
radware/vdirect_user = vDirect	(StrOpt) vdirect user name
ssl/ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
ssl/cert_file = None	(StrOpt) Certificate file to use when starting the server securely

Option = default value ssl/key_file = None	(Type) Help string (StrOpt) Private key file to use when starting the server securely
--	---

Table 7.66. New default values

Option	Previous default value	New default value
allowed_rpc_exception_modules	neutron.openstack.common.exception.nova.exception, cinder.exception, exceptions	nova.exception, cinder.exception, exceptions
default_log_levels	amqp=WARN, amqpplib=WARN, boto=WARN, eventlet.wsgi.server=WARN, keystone=INFO, qpid=WARN, sqlalchemy=WARN, suds=INFO	amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN
device_driver	neutron.services.loadbalancer.drivers.haproxynamespace_driver	haproxy.namespace_driver
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%request_id)s %(user_identity)s] %(instance)s%(message)s
AGENT/minimize_polling	False	True
NOVA/vif_types	unbound, binding_failed, ovs, ivs, bridge, 802.1qbg, 802.1qbh, hyperv, other	unbound, binding_failed, ovs, ivs, bridge, 802.1qbg, 802.1qbh, hyperv, midonet, mlnx_direct, hostdev, other
RESTPROXY/server_auth	username:password	None
RESTPROXY/server_ssl	False	True
RESTPROXY/servers	localhost:8800	localhost:8800
SECURITYGROUP/firewall_driver	neutron.agent.firewall.NoopFirewallDriver	None
fwaas/driver	neutron.services.firewall.drivers.fwaas_base.NoopFwaasDriver	base.NoopFwaasDriver

Table 7.67. Deprecated options

Deprecated option	New Option
nvp_controllers	nsx_controllers
NVP/nvp_gen_timeout	NSX/nsx_gen_timeout
dnsmasq_dns_server	dnsmasq_dns_servers
log_config	log_config_append
nvp_password	nsx_password
loadbalancer_state_path	haproxy/loadbalancer_state_path
user_group	haproxy/user_group
nvp_user	nsx_user

8. Object Storage

Table of Contents

Introduction to Object Storage	425
Object Storage general service configuration	425
Object server configuration	427
Object expirer configuration	435
Container server configuration	438
Account server configuration	444
Proxy server configuration	450
Proxy server memcache configuration	466
Rsyncd configuration	466
Configure Object Storage features	467

OpenStack Object Storage uses multiple configuration files for multiple services and background daemons, and **paste.deploy** to manage server configurations. Default configuration options appear in the [DEFAULT] section. You can override the default values by setting values in the other sections.

Introduction to Object Storage

Object Storage is a robust, highly scalable and fault tolerant storage platform for unstructured data such as objects. Objects are stored bits, accessed through a RESTful, HTTP-based interface. You cannot access data at the block or file level. Object Storage is commonly used to archive and back up data, with use cases in virtual machine image, photo, video and music storage.

Object Storage provides a high degree of availability, throughput, and performance with its scale out architecture. Each object is replicated across multiple servers, residing within the same data center or across data centers, which mitigates the risk of network and hardware failure. In the event of hardware failure, Object Storage will automatically copy objects to a new location to ensure that there are always three copies available. Object Storage is an eventually consistent distributed storage platform; it sacrifices consistency for maximum availability and partition tolerance. Object Storage enables you to create a reliable platform by using commodity hardware and inexpensive storage.

For more information, review the key concepts in the developer documentation at docs.openstack.org/developer/swift/.

Object Storage general service configuration

Most Object Storage services fall into two categories, Object Storage's wsgi servers and background daemons.

Object Storage uses **paste.deploy** to manage server configurations. Read more at <http://pythonpaste.org/deploy/>.

Default configuration options are set in the `[DEFAULT]` section, and any options specified there can be overridden in any of the other sections when the syntax `set option_name = value` is in place.

Configuration for servers and daemons can be expressed together in the same file for each type of server, or separately. If a required section for the service trying to start is missing there will be an error. Sections not used by the service are ignored.

Consider the example of an Object Storage node. By convention configuration for the `object-server`, `object-updater`, `object-replicator`, and `object-auditor` exist in a single file `/etc/swift/object-server.conf`:

```
[DEFAULT]

[pipeline:main]
pipeline = object-server

[app:object-server]
use = egg:swift#object

[object-replicator]
reclaim_age = 259200

[object-updater]

[object-auditor]
```

Object Storage services expect a configuration path as the first argument:

```
$ swift-object-auditor
Usage: swift-object-auditor CONFIG [options]

Error: missing config path argument
```

If you omit the `object-auditor` section, this file cannot be used as the configuration path when starting the `swift-object-auditor` daemon:

```
$ swift-object-auditor /etc/swift/object-server.conf
Unable to find object-auditor config section in /etc/swift/object-server.conf
```

If the configuration path is a directory instead of a file all of the files in the directory with the file extension ".conf" will be combined to generate the configuration object which is delivered to the Object Storage service. This is referred to generally as "directory-based configuration".

Directory-based configuration leverages ConfigParser's native multi-file support. Files ending in ".conf" in the given directory are parsed in lexicographical order. File names starting with "." are ignored. A mixture of file and directory configuration paths is not supported - if the configuration path is a file, only that file will be parsed.

The Object Storage service management tool `swift-init` has adopted the convention of looking for `/etc/swift/{type}-server.conf.d/` if the file `/etc/swift/{type}-server.conf` file does not exist.

When using directory-based configuration, if the same option under the same section appears more than once in different files, the last value parsed is said to override previous

occurrences. You can ensure proper override precedence by prefixing the files in the configuration directory with numerical values, as in the following example file layout:

```
/etc/swift/
  default.base
  object-server.conf.d/
    000_default.conf -> ../default.base
    001_default-override.conf
    010_server.conf
    020_replicator.conf
    030_updater.conf
    040_auditor.conf
```

You can inspect the resulting combined configuration object using the **swift-config** command-line tool.

All the services of an Object Store deployment share a common configuration in the `[swift-hash]` section of the `/etc/swift/swift.conf` file. The `swift_hash_path_suffix` and `swift_hash_path_prefix` values must be identical on all the nodes.

Table 8.1. Description of configuration options for `[swift-hash]` in `swift.conf-sample`

Configuration option = Default value	Description
<code>swift_hash_path_suffix = changeme</code>	A suffix used by <code>hash_path</code> to offer a bit more security when generating hashes for paths. It simply appends this value to all paths; if someone knows this suffix, it's easier for them to guess the hash a path will end up with. New installations are advised to set this parameter to a random secret, which would not be disclosed outside the organization. The same secret needs to be used by all swift servers of the same cluster. Existing installations should set this parameter to an empty string.
<code>swift_hash_path_prefix = changeme</code>	A prefix used by <code>hash_path</code> to offer a bit more security when generating hashes for paths. It simply appends this value to all paths; if someone knows this suffix, it's easier for them to guess the hash a path will end up with. New installations are advised to set this parameter to a random secret, which would not be disclosed outside the organization. The same secret needs to be used by all swift servers of the same cluster. Existing installations should set this parameter to an empty string.

Object server configuration

Find an example object server configuration at `etc/object-server.conf-sample` in the source code repository.

The available configuration options are:

Table 8.2. Description of configuration options for `[DEFAULT]` in `object-server.conf-sample`

Configuration option = Default value	Description
<code>bind_ip = 0.0.0.0</code>	IP Address for server to bind to
<code>bind_port = 6000</code>	Port for server to bind to

Configuration option = Default value	Description
bind_timeout = 30	Seconds to attempt bind before giving up
backlog = 4096	Maximum number of allowed pending TCP connections
user = swift	User to run as
swift_dir = /etc/swift	Swift configuration directory
devices = /srv/node	Parent directory of where devices are mounted
mount_check = true	Whether or not check if the devices are mounted to prevent accidentally writing to the root device
disable_fallocate = false	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
expiring_objects_container_divisor = 86400	No help text available for this option.
expiring_objects_account_name = expiring_objects	No help text available for this option.
workers = auto	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.
max_clients = 1024	Maximum number of clients one worker can process simultaneously Lowering the number of clients handled per worker, and raising the number of workers can lessen the impact that a CPU intensive, or blocking, request can have on other requests served by the same worker. If the maximum number of clients is set to one, then a given worker will not perform another call while processing, allowing other workers a chance to process it.
log_name = swift	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_udp_host =	If not set, the UDB receiver for syslog is disabled.
log_udp_port = 514	Port value for UDB receiver, if enabled.
log_statsd_host = localhost	If not set, the StatsD feature is disabled.
log_statsd_port = 8125	Port value for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_sample_rate_factor = 1.0	Not recommended to set this to a value less than 1.0, if frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
eventlet_debug = false	If true, turn on debug logging for eventlet
fallocate_reserve = 0	You can set fallocate_reserve to the number of bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. server. For most cases, this should be `egg:swift#object`.
conn_timeout = 0.5	Connection timeout to external services
node_timeout = 3	Request timeout to external services
client_timeout = 60	Timeout to read one chunk from a client external services
network_chunk_size = 65536	Size of chunks to read/write over the network
disk_chunk_size = 65536	Size of chunks to read/write to disk

Table 8.3. Description of configuration options for [app:object-server] in object-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#object	Entry point of paste.deploy in the server
set log_name = object-server	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_requests = true	Whether or not to log requests
set log_address = /dev/log	Location where syslog sends the logs to
max_upload_time = 86400	Maximum time allowed to upload an object
slow = 0	If > 0, Minimum time in seconds for a PUT or DELETE request to complete
keep_cache_size = 5424880	Largest object size to keep in buffer cache
keep_cache_private = false	Allow non-public objects to stay in kernel's buffer cache
mb_per_sync = 512	On PUT requests, sync file every n MB
allowed_headers = Content-Disposition, Content-Encoding, X-Delete-At, X-Object-Manifest, X-Static-Large-Object	Comma-separated list of headers that can be set in metadata of an object
auto_create_account_prefix = .	Prefix to use when automatically creating accounts
threads_per_disk = 0	Size of the per-disk thread pool used for performing disk I/O. The default of 0 means to not use a per-disk thread pool. It is recommended to keep this value small, as large values can result in high read latencies due to large queue depths. A good starting point is 4 threads per disk.
replication_server = false	If defined, tells server how to handle replication verbs in requests. When set to True (or 1), only replication verbs will be accepted. When set to False, replication verbs will be rejected. When undefined, server will accept any verb in the request.
replication_concurrency = 4	Set to restrict the number of concurrent incoming REPLICATION requests; set to 0 for unlimited
replication_one_per_device = True	Restricts incoming REPLICATION requests to one per device, replication_concurrency above allowing. This can help control I/O to each device, but you may wish to set this to False to allow multiple REPLICATION requests (up to the above replication_concurrency setting) per device.
replication_lock_timeout = 15	Number of seconds to wait for an existing replication device lock before giving up.
replication_failure_threshold = 100	The number of subrequest failures before the replication_failure_ratio is checked
replication_failure_ratio = 1.0	If the value of failures / successes of REPLICATION subrequests exceeds this ratio, the overall REPLICATION request will be aborted

Table 8.4. Description of configuration options for [pipeline:main] in object-server.conf-sample

Configuration option = Default value	Description
pipeline = healthcheck recon object-server	No help text available for this option.

Table 8.5. Description of configuration options for [object-replicator] in object-server.conf-sample

Configuration option = Default value	Description
log_name = object-replicator	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
vm_test_mode = no	Indicates that you are using a VM environment
daemonize = on	Whether or not to run replication as a daemon
run_pause = 30	Time in seconds to wait between replication passes
concurrency = 1	Number of replication workers to spawn
stats_interval = 300	Interval in seconds between logging replication statistics
sync_method = rsync	No help text available for this option.
rsync_timeout = 900	Max duration (seconds) of a partition rsync
rsync_bwlimit = 0	No help text available for this option.
rsync_io_timeout = 30	Passed to rsync for a max duration (seconds) of an I/O op
node_timeout = <whatever's in the DEFAULT section or 10>	Request timeout to external services
http_timeout = 60	Maximum duration for an HTTP request
lockup_timeout = 1800	Attempts to kill all workers if nothing replications for lockup_timeout seconds
reclaim_age = 604800	Time elapsed in seconds before an object can be reclaimed
ring_check_interval = 15	How often (in seconds) to check the ring
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored
rsync_error_log_line_length = 0	No help text available for this option.
handoffs_first = False	If set to True, partitions that are not supposed to be on the node will be replicated first. The default setting should not be changed, except for extreme situations.
handoff_delete = auto	By default handoff partitions will be removed when it has successfully replicated to all the canonical nodes. If set to an integer n, it will remove the partition if it is successfully replicated to n nodes. The default setting should not be changed, except for extremem situations. This uses what's set here, or what's set in the DEFAULT section, or 10 (though other sections use 3 as the final default).

Table 8.6. Description of configuration options for [object-updater] in object-server.conf-sample

Configuration option = Default value	Description
log_name = object-updater	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
interval = 300	Minimum time for a pass to take
concurrency = 1	Number of replication workers to spawn
node_timeout = <whatever's in the DEFAULT section or 10>	Request timeout to external services
slowdown = 0.01	Time in seconds to wait between objects
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.7. Description of configuration options for [object-auditor] in object-server.conf-sample

Configuration option = Default value	Description
log_name = object-auditor	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
files_per_second = 20	Maximum files audited per second. Should be tuned according to individual system specs. 0 is unlimited.
bytes_per_second = 10000000	Maximum bytes audited per second. Should be tuned according to individual system specs. 0 is unlimited. mounted to prevent accidentally writing to the root device process simultaneously (it will actually accept(2) N + 1). Setting this to one (1) will only handle one request at a time, without accepting another request concurrently. By increasing the number of workers to a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests. underlying filesystem does not support it. to setup custom log handlers. bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. container server. For most cases, this should be `egg:swift#container`.
log_time = 3600	Frequency of status logs in seconds.
zero_byte_files_per_second = 50	Maximum zero byte files audited per second.
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored
object_size_stats =	No help text available for this option.

Table 8.8. Description of configuration options for [filter:healthcheck] in object-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#healthcheck	Entry point of paste.deploy in the server
disable_path =	No help text available for this option.

Table 8.9. Description of configuration options for [filter:recon] in object-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#recon	Entry point of paste.deploy in the server
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored
recon_lock_path = /var/lock	No help text available for this option.

Sample object server configuration file

```
[DEFAULT]
# bind_ip = 0.0.0.0
# bind_port = 6000
# bind_timeout = 30
# backlog = 4096
# user = swift
# swift_dir = /etc/swift
```



```
# set log_level = INFO
# set log_requests = true
# set log_address = /dev/log
#
# max_upload_time = 86400
# slow = 0
#
# Objects smaller than this are not evicted from the buffercache once read
# keep_cache_size = 5424880
#
# If true, objects for authenticated GET requests may be kept in buffer cache
# if small enough
# keep_cache_private = false
#
# on PUTs, sync data every n MB
# mb_per_sync = 512
#
# Comma separated list of headers that can be set in metadata on an object.
# This list is in addition to X-Object-Meta-* headers and cannot include
# Content-Type, etag, Content-Length, or deleted
# allowed_headers = Content-Disposition, Content-Encoding, X-Delete-At, X-
Object-Manifest, X-Static-Large-Object
#
# auto_create_account_prefix = .
#
# A value of 0 means "don't use thread pools". A reasonable starting point is
# 4.
# threads_per_disk = 0
#
# Configure parameter for creating specific server
# To handle all verbs, including replication verbs, do not specify
# "replication_server" (this is the default). To only handle replication,
# set to a True value (e.g. "True" or "1"). To handle only non-replication
# verbs, set to "False". Unless you have a separate replication network, you
# should not specify any value for "replication_server".
# replication_server = false
#
# Set to restrict the number of concurrent incoming REPLICATION requests
# Set to 0 for unlimited
# Note that REPLICATION is currently an ssync only item
# replication_concurrency = 4
#
# Restricts incoming REPLICATION requests to one per device,
# replication_currency above allowing. This can help control I/O to each
# device, but you may wish to set this to False to allow multiple REPLICATION
# requests (up to the above replication_concurrency setting) per device.
# replication_one_per_device = True
#
# Number of seconds to wait for an existing replication device lock before
# giving up.
# replication_lock_timeout = 15
#
# These next two settings control when the REPLICATION subrequest handler will
# abort an incoming REPLICATION attempt. An abort will occur if there are at
# least threshold number of failures and the value of failures / successes
# exceeds the ratio. The defaults of 100 and 1.0 means that at least 100
# failures have to occur and there have to be more failures than successes for
# an abort to occur.
# replication_failure_threshold = 100
# replication_failure_ratio = 1.0
```

```
[filter:healthcheck]
use = egg:swift#healthcheck
# An optional filesystem path, which if present, will cause the healthcheck
# URL to return "503 Service Unavailable" with a body of "DISABLED BY FILE"
# disable_path =

[filter:recon]
use = egg:swift#recon
#recon_cache_path = /var/cache/swift
#recon_lock_path = /var/lock

[object-replicator]
# You can override the default log routing for this app here (don't use set!):
# log_name = object-replicator
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# vm_test_mode = no
# daemonize = on
# run_pause = 30
# concurrency = 1
# stats_interval = 300
#
# The sync method to use; default is rsync but you can use ssync to try the
# EXPERIMENTAL all-swift-code-no-rsync-callouts method. Once ssync is verified
# as having performance comparable to, or better than, rsync, we plan to
# deprecate rsync so we can move on with more features for replication.
# sync_method = rsync
#
# max duration of a partition rsync
# rsync_timeout = 900
#
# bandwidth limit for rsync in kB/s. 0 means unlimited
# rsync_bwlimit = 0
#
# passed to rsync for io op timeout
# rsync_io_timeout = 30
#
# node_timeout = <whatever's in the DEFAULT section or 10>
# max duration of an http request; this is for REPLICATE finalization calls
# and
# so should be longer than node_timeout
# http_timeout = 60
#
# attempts to kill all workers if nothing replicates for lockup_timeout
# seconds
# lockup_timeout = 1800
#
# The replicator also performs reclamation
# reclaim_age = 604800
#
# ring_check_interval = 15
# recon_cache_path = /var/cache/swift
#
# limits how long rsync error log lines are
# 0 means to log the entire line
# rsync_error_log_line_length = 0
#
```


Table 8.10. Description of configuration options for [DEFAULT] in object-expirer.conf-sample

Configuration option = Default value	Description
swift_dir = /etc/swift	Swift configuration directory
user = swift	User to run as
log_name = swift	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_udp_host =	If not set, the UDB receiver for syslog is disabled.
log_udp_port = 514	Port value for UDB receiver, if enabled.
log_statsd_host = localhost	If not set, the StatsD feature is disabled.
log_statsd_port = 8125	Port value for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_sample_rate_factor = 1.0	Not recommended to set this to a value less than 1.0, if frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.

Table 8.11. Description of configuration options for [app:proxy-server] in object-expirer.conf-sample

Configuration option = Default value	Description
use = egg:swift#proxy	Entry point of paste.deploy in the server

Table 8.12. Description of configuration options for [filter:cache] in object-expirer.conf-sample

Configuration option = Default value	Description
use = egg:swift#memcache	Entry point of paste.deploy in the server

Table 8.13. Description of configuration options for [filter:catch_errors] in object-expirer.conf-sample

Configuration option = Default value	Description
use = egg:swift#catch_errors	Entry point of paste.deploy in the server

Table 8.14. Description of configuration options for [object-expirer] in object-expirer.conf-sample

Configuration option = Default value	Description
interval = 300	Minimum time for a pass to take
auto_create_account_prefix = .	Prefix to use when automatically creating accounts
expiring_objects_account_name = expiring_objects	No help text available for this option.
report_interval = 300	No help text available for this option.

Configuration option = Default value	Description
concurrency = 1	Number of replication workers to spawn
processes = 0	No help text available for this option.
process = 0	(it will actually accept(2) N + 1). Setting this to one (1) will only handle one request at a time, without accepting another request concurrently.

Table 8.15. Description of configuration options for [pipeline:main] in object-expirer.conf-sample

Configuration option = Default value	Description
pipeline = catch_errors cache proxy-server	No help text available for this option.

Sample object expirer configuration file

```
[DEFAULT]
# swift_dir = /etc/swift
# user = swift
# You can specify default log routing here if you want:
# log_name = swift
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# comma separated list of functions to call to setup custom log handlers.
# functions get passed: conf, name, log_to_console, log_route, fmt, logger,
# adapted_logger
# log_custom_handlers =
#
# If set, log_udp_host will override log_address
# log_udp_host =
# log_udp_port = 514
#
# You can enable StatsD logging here:
# log_statsd_host = localhost
# log_statsd_port = 8125
# log_statsd_default_sample_rate = 1.0
# log_statsd_sample_rate_factor = 1.0
# log_statsd_metric_prefix =

[object-expirer]
# interval = 300
# auto_create_account_prefix = .
# expiring_objects_account_name = expiring_objects
# report_interval = 300
# concurrency is the level of concurrency o use to do the work, this value
# must be set to at least 1
# concurrency = 1
# processes is how many parts to divide the work into, one part per process
# that will be doing the work
# processes set 0 means that a single process will be doing all the work
# processes can also be specified on the command line and will override the
# config value
# processes = 0
# process is which of the parts a particular process will work on
# process can also be specified on the command line and will override the
# config
# value
```



```
# process is "zero based", if you want to use 3 processes, you should run
# processes with process set to 0, 1, and 2
# process = 0

[pipeline:main]
pipeline = catch_errors cache proxy-server

[app:proxy-server]
use = egg:swift#proxy
# See proxy-server.conf-sample for options

[filter:cache]
use = egg:swift#memcache
# See proxy-server.conf-sample for options

[filter:catch_errors]
use = egg:swift#catch_errors
# See proxy-server.conf-sample for options
```

Container server configuration

Find an example container server configuration at `etc/container-server.conf-sample` in the source code repository.

The available configuration options are:

Table 8.16. Description of configuration options for [DEFAULT] in `container-server.conf-sample`

Configuration option = Default value	Description
<code>bind_ip = 0.0.0.0</code>	IP Address for server to bind to
<code>bind_port = 6001</code>	Port for server to bind to
<code>bind_timeout = 30</code>	Seconds to attempt bind before giving up
<code>backlog = 4096</code>	Maximum number of allowed pending TCP connections
<code>user = swift</code>	User to run as
<code>swift_dir = /etc/swift</code>	Swift configuration directory
<code>devices = /srv/node</code>	Parent directory of where devices are mounted
<code>mount_check = true</code>	Whether or not check if the devices are mounted to prevent accidentally writing to the root device
<code>disable_fallocate = false</code>	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
<code>workers = auto</code>	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.
<code>max_clients = 1024</code>	Maximum number of clients one worker can process simultaneously Lowering the number of clients handled per worker, and raising the number of workers can lessen the impact that a CPU intensive, or blocking, request can have on other requests served by the same worker. If the maximum number of clients is set to one, then a given worker will not perform another call while processing, allowing other workers a chance to process it.
<code>allowed_sync_hosts = 127.0.0.1</code>	No help text available for this option.
<code>log_name = swift</code>	Label used when logging
<code>log_facility = LOG_LOCAL0</code>	Syslog log facility

Configuration option = Default value	Description
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_udp_host =	If not set, the UDB receiver for syslog is disabled.
log_udp_port = 514	Port value for UDB receiver, if enabled.
log_statsd_host = localhost	If not set, the StatsD feature is disabled.
log_statsd_port = 8125	Port value for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_sample_rate_factor = 1.0	Not recommended to set this to a value less than 1.0, if frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
db_preallocation = off	If you don't mind the extra disk space usage in overhead, you can turn this on to preallocate disk space with SQLite databases to decrease fragmentation. underlying filesystem does not support it. to setup custom log handlers. bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. server. For most cases, this should be `egg:swift#account`. replication passes account can be reclaimed
eventlet_debug = false	If true, turn on debug logging for eventlet
fallocate_reserve = 0	You can set fallocate_reserve to the number of bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. server. For most cases, this should be `egg:swift#object`.

Table 8.17. Description of configuration options for [app:container-server] in container-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#container	Entry point of paste.deploy in the server
set log_name = container-server	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_requests = true	Whether or not to log requests
set log_address = /dev/log	Location where syslog sends the logs to
node_timeout = 3	Request timeout to external services
conn_timeout = 0.5	Connection timeout to external services
allow_versions = false	Enable/Disable object versioning feature
auto_create_account_prefix = .	Prefix to use when automatically creating accounts
replication_server = false	If defined, tells server how to handle replication verbs in requests. When set to True (or 1), only replication verbs will be accepted. When set to False, replication verbs will be rejected. When undefined, server will accept any verb in the request.

Table 8.18. Description of configuration options for [pipeline:main] in container-server.conf-sample

Configuration option = Default value	Description
pipeline = healthcheck recon container-server	No help text available for this option.

Table 8.19. Description of configuration options for [container-replicator] in container-server.conf-sample

Configuration option = Default value	Description
log_name = container-replicator	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
vm_test_mode = no	Indicates that you are using a VM environment
per_diff = 1000	Limit number of items to get per diff
max_diffs = 100	Caps how long the replicator spends trying to sync a database per pass
concurrency = 8	Number of replication workers to spawn
interval = 30	Minimum time for a pass to take
node_timeout = 10	Request timeout to external services
conn_timeout = 0.5	Connection timeout to external services
reclaim_age = 604800	Time elapsed in seconds before an object can be reclaimed
run_pause = 30	Time in seconds to wait between replication passes
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.20. Description of configuration options for [container-updater] in container-server.conf-sample

Configuration option = Default value	Description
log_name = container-updater	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
interval = 300	Minimum time for a pass to take
concurrency = 4	Number of replication workers to spawn
node_timeout = 3	Request timeout to external services
conn_timeout = 0.5	Connection timeout to external services
slowdown = 0.01	Time in seconds to wait between objects
account_suppression_time = 60	Seconds to suppress updating an account that has generated an error (timeout, not yet found, etc.)
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.21. Description of configuration options for [container-auditor] in container-server.conf-sample

Configuration option = Default value	Description
log_name = container-auditor	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level

Configuration option = Default value	Description
log_address = /dev/log	Location where syslog sends the logs to
interval = 1800	Minimum time for a pass to take
containers_per_second = 200	Maximum containers audited per second. Should be tuned according to individual system specs. 0 is unlimited. mounted to prevent accidentally writing to the root device process simultaneously (it will actually accept(2) N + 1). Setting this to one (1) will only handle one request at a time, without accepting another request concurrently. By increasing the number of workers to a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.22. Description of configuration options for [container-sync] in container-server.conf-sample

Configuration option = Default value	Description
log_name = container-sync	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
sync_proxy = http://10.1.1.1:8888,http://10.1.1.2:8888	If you need to use an HTTP proxy, set it here. Defaults to no proxy.
interval = 300	Minimum time for a pass to take
container_time = 60	Maximum amount of time to spend syncing each container

Table 8.23. Description of configuration options for [filter:healthcheck] in container-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#healthcheck	Entry point of paste.deploy in the server
disable_path =	No help text available for this option.

Table 8.24. Description of configuration options for [filter:recon] in container-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#recon	Entry point of paste.deploy in the server
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Sample container server configuration file

```
[DEFAULT]
# bind_ip = 0.0.0.0
# bind_port = 6001
# bind_timeout = 30
# backlog = 4096
# user = swift
# swift_dir = /etc/swift
# devices = /srv/node
# mount_check = true
# disable_fallocate = false
#
```

```
# Use an integer to override the number of pre-forked processes that will
# accept connections.
# workers = auto
#
# Maximum concurrent requests per worker
# max_clients = 1024
#
# This is a comma separated list of hosts allowed in the X-Container-Sync-To
# field for containers. This is the old-style of using container sync. It is
# strongly recommended to use the new style of a separate
# container-sync-realms.conf -- see container-sync-realms.conf-sample
# allowed_sync_hosts = 127.0.0.1
#
# You can specify default log routing here if you want:
# log_name = swift
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# comma separated list of functions to call to setup custom log handlers.
# functions get passed: conf, name, log_to_console, log_route, fmt, logger,
# adapted_logger
# log_custom_handlers =
#
# If set, log_udp_host will override log_address
# log_udp_host =
# log_udp_port = 514
#
# You can enable StatsD logging here:
# log_statsd_host = localhost
# log_statsd_port = 8125
# log_statsd_default_sample_rate = 1.0
# log_statsd_sample_rate_factor = 1.0
# log_statsd_metric_prefix =
#
# If you don't mind the extra disk space usage in overhead, you can turn this
# on to preallocate disk space with SQLite databases to decrease
# fragmentation.
# db_preallocation = off
#
# eventlet_debug = false
#
# You can set fallocate_reserve to the number of bytes you'd like fallocate to
# reserve, whether there is space for the given file size or not.
# fallocate_reserve = 0

[pipeline:main]
pipeline = healthcheck recon container-server

[app:container-server]
use = egg:swift#container
# You can override the default log routing for this app here:
# set log_name = container-server
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_requests = true
# set log_address = /dev/log
#
# node_timeout = 3
# conn_timeout = 0.5
```

```
# allow_versions = false
# auto_create_account_prefix = .
#
# Configure parameter for creating specific server
# To handle all verbs, including replication verbs, do not specify
# "replication_server" (this is the default). To only handle replication,
# set to a True value (e.g. "True" or "1"). To handle only non-replication
# verbs, set to "False". Unless you have a separate replication network, you
# should not specify any value for "replication_server".
# replication_server = false

[filter:healthcheck]
use = egg:swift#healthcheck
# An optional filesystem path, which if present, will cause the healthcheck
# URL to return "503 Service Unavailable" with a body of "DISABLED BY FILE"
# disable_path =

[filter:recon]
use = egg:swift#recon
#recon_cache_path = /var/cache/swift

[container-replicator]
# You can override the default log routing for this app here (don't use set!):
# log_name = container-replicator
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# vm_test_mode = no
# per_diff = 1000
# max_diffs = 100
# concurrency = 8
# interval = 30
# node_timeout = 10
# conn_timeout = 0.5
#
# The replicator also performs reclamation
# reclaim_age = 604800
#
# Time in seconds to wait between replication passes
# run_pause = 30
#
# recon_cache_path = /var/cache/swift

[container-updater]
# You can override the default log routing for this app here (don't use set!):
# log_name = container-updater
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# interval = 300
# concurrency = 4
# node_timeout = 3
# conn_timeout = 0.5
#
# slowdown will sleep that amount between containers
# slowdown = 0.01
#
# Seconds to suppress updating an account that has generated an error
```

```
# account_suppression_time = 60
#
# recon_cache_path = /var/cache/swift

[container-auditor]
# You can override the default log routing for this app here (don't use set!):
# log_name = container-auditor
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# Will audit each container at most once per interval
# interval = 1800
#
# containers_per_second = 200
# recon_cache_path = /var/cache/swift

[container-sync]
# You can override the default log routing for this app here (don't use set!):
# log_name = container-sync
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# If you need to use an HTTP Proxy, set it here; defaults to no proxy.
# You can also set this to a comma separated list of HTTP Proxies and they
will
# be randomly used (simple load balancing).
# sync_proxy = http://10.1.1.1:8888,http://10.1.1.2:8888
#
# Will sync each container at most once per interval
# interval = 300
#
# Maximum amount of time to spend syncing each container per pass
# container_time = 60
```

Account server configuration

Find an example account server configuration at `etc/account-server.conf-sample` in the source code repository.

The available configuration options are:

Table 8.25. Description of configuration options for [DEFAULT] in `account-server.conf-sample`

Configuration option = Default value	Description
<code>bind_ip = 0.0.0.0</code>	IP Address for server to bind to
<code>bind_port = 6002</code>	Port for server to bind to
<code>bind_timeout = 30</code>	Seconds to attempt bind before giving up
<code>backlog = 4096</code>	Maximum number of allowed pending TCP connections
<code>user = swift</code>	User to run as
<code>swift_dir = /etc/swift</code>	Swift configuration directory
<code>devices = /srv/node</code>	Parent directory of where devices are mounted
<code>mount_check = true</code>	Whether or not check if the devices are mounted to prevent accidentally writing to the root device

Configuration option = Default value	Description
disable_fallocate = false	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
workers = auto	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.
max_clients = 1024	Maximum number of clients one worker can process simultaneously Lowering the number of clients handled per worker, and raising the number of workers can lessen the impact that a CPU intensive, or blocking, request can have on other requests served by the same worker. If the maximum number of clients is set to one, then a given worker will not perform another call while processing, allowing other workers a chance to process it.
log_name = swift	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_udp_host =	If not set, the UDB receiver for syslog is disabled.
log_udp_port = 514	Port value for UDB receiver, if enabled.
log_statsd_host = localhost	If not set, the StatsD feature is disabled.
log_statsd_port = 8125	Port value for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_sample_rate_factor = 1.0	Not recommended to set this to a value less than 1.0, if frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
db_preallocation = off	If you don't mind the extra disk space usage in overhead, you can turn this on to preallocate disk space with SQLite databases to decrease fragmentation. underlying filesystem does not support it. to setup custom log handlers. bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. server. For most cases, this should be `egg:swift#account`. replication passes account can be reclaimed
eventlet_debug = false	If true, turn on debug logging for eventlet
fallocate_reserve = 0	You can set fallocate_reserve to the number of bytes you'd like fallocate to reserve, whether there is space for the given file size or not. This is useful for systems that behave badly when they completely run out of space; you can make the services pretend they're out of space early. server. For most cases, this should be `egg:swift#object`.

Table 8.26. Description of configuration options for [app:account-server] in account-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#account	Entry point of paste.deploy in the server
set log_name = account-server	Label to use when logging

Configuration option = Default value	Description
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_requests = true	Whether or not to log requests
set log_address = /dev/log	Location where syslog sends the logs to
auto_create_account_prefix = .	Prefix to use when automatically creating accounts
replication_server = false	If defined, tells server how to handle replication verbs in requests. When set to True (or 1), only replication verbs will be accepted. When set to False, replication verbs will be rejected. When undefined, server will accept any verb in the request.

Table 8.27. Description of configuration options for [pipeline:main] in account-server.conf-sample

Configuration option = Default value	Description
pipeline = healthcheck recon account-server	No help text available for this option.

Table 8.28. Description of configuration options for [account-replicator] in account-server.conf-sample

Configuration option = Default value	Description
log_name = account-replicator	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
vm_test_mode = no	Indicates that you are using a VM environment
per_diff = 1000	Limit number of items to get per diff
max_diffs = 100	Caps how long the replicator spends trying to sync a database per pass
concurrency = 8	Number of replication workers to spawn
interval = 30	Minimum time for a pass to take
error_suppression_interval = 60	Time in seconds that must elapse since the last error for a node to be considered no longer error limited
error_suppression_limit = 10	Error count to consider a node error limited
node_timeout = 10	Request timeout to external services
conn_timeout = 0.5	Connection timeout to external services
reclaim_age = 604800	Time elapsed in seconds before an object can be reclaimed
run_pause = 30	Time in seconds to wait between replication passes
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.29. Description of configuration options for [account-auditor] in account-server.conf-sample

Configuration option = Default value	Description
log_name = account-auditor	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
interval = 1800	Minimum time for a pass to take

Configuration option = Default value	Description
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
accounts_per_second = 200	Maximum accounts audited per second. Should be tuned according to individual system specs. 0 is unlimited.
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 8.30. Description of configuration options for [account-reaper] in account-server.conf-sample

Configuration option = Default value	Description
log_name = account-reaper	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_address = /dev/log	Location where syslog sends the logs to
concurrency = 25	Number of replication workers to spawn
interval = 3600	Minimum time for a pass to take
node_timeout = 10	Request timeout to external services
conn_timeout = 0.5	Connection timeout to external services
delay_reaping = 0	Normally, the reaper begins deleting account information for deleted accounts immediately; you can set this to delay its work however. The value is in seconds, 2592000 = 30 days, for example. bind to giving up worker can process simultaneously (it will actually accept(2) N + 1). Setting this to one (1) will only handle one request at a time, without accepting another request concurrently. By increasing the number of workers to a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.
reap_warn_after = 2592000	No help text available for this option.

Table 8.31. Description of configuration options for [filter:healthcheck] in account-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#healthcheck	Entry point of paste.deploy in the server
disable_path =	No help text available for this option.

Table 8.32. Description of configuration options for [filter:recon] in account-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#recon	Entry point of paste.deploy in the server
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Sample account server configuration file

```
[DEFAULT]
# bind_ip = 0.0.0.0
# bind_port = 6002
# bind_timeout = 30
# backlog = 4096
# user = swift
# swift_dir = /etc/swift
```

```
# devices = /srv/node
# mount_check = true
# disable_fallocate = false
#
# Use an integer to override the number of pre-forked processes that will
# accept connections.
# workers = auto
#
# Maximum concurrent requests per worker
# max_clients = 1024
#
# You can specify default log routing here if you want:
# log_name = swift
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# comma separated list of functions to call to setup custom log handlers.
# functions get passed: conf, name, log_to_console, log_route, fmt, logger,
# adapted_logger
# log_custom_handlers =
#
# If set, log_udp_host will override log_address
# log_udp_host =
# log_udp_port = 514
#
# You can enable StatsD logging here:
# log_statsd_host = localhost
# log_statsd_port = 8125
# log_statsd_default_sample_rate = 1.0
# log_statsd_sample_rate_factor = 1.0
# log_statsd_metric_prefix =
#
# If you don't mind the extra disk space usage in overhead, you can turn this
# on to preallocate disk space with SQLite databases to decrease
# fragmentation.
# db_preallocation = off
#
# eventlet_debug = false
#
# You can set fallocate_reserve to the number of bytes you'd like fallocate to
# reserve, whether there is space for the given file size or not.
# fallocate_reserve = 0

[pipeline:main]
pipeline = healthcheck recon account-server

[app:account-server]
use = egg:swift#account
# You can override the default log routing for this app here:
# set log_name = account-server
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_requests = true
# set log_address = /dev/log
#
# auto_create_account_prefix = .
#
# Configure parameter for creating specific server
# To handle all verbs, including replication verbs, do not specify
```

```
# "replication_server" (this is the default). To only handle replication,
# set to a True value (e.g. "True" or "1"). To handle only non-replication
# verbs, set to "False". Unless you have a separate replication network, you
# should not specify any value for "replication_server".
# replication_server = false

[filter:healthcheck]
use = egg:swift#healthcheck
# An optional filesystem path, which if present, will cause the healthcheck
# URL to return "503 Service Unavailable" with a body of "DISABLED BY FILE"
# disable_path =

[filter:recon]
use = egg:swift#recon
# recon_cache_path = /var/cache/swift

[account-replicator]
# You can override the default log routing for this app here (don't use set!):
# log_name = account-replicator
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# vm_test_mode = no
# per_diff = 1000
# max_diffs = 100
# concurrency = 8
# interval = 30
#
# How long without an error before a node's error count is reset. This will
# also be how long before a node is reenabled after suppression is triggered.
# error_suppression_interval = 60
#
# How many errors can accumulate before a node is temporarily ignored.
# error_suppression_limit = 10
#
# node_timeout = 10
# conn_timeout = 0.5
#
# The replicator also performs reclamation
# reclaim_age = 604800
#
# Time in seconds to wait between replication passes
# run_pause = 30
#
# recon_cache_path = /var/cache/swift

[account-auditor]
# You can override the default log routing for this app here (don't use set!):
# log_name = account-auditor
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# Will audit each account at most once per interval
# interval = 1800
#
# log_facility = LOG_LOCAL0
# log_level = INFO
# accounts_per_second = 200
```


Configuration option = Default value	Description
max_clients = 1024	Maximum number of clients one worker can process simultaneously Lowering the number of clients handled per worker, and raising the number of workers can lessen the impact that a CPU intensive, or blocking, request can have on other requests served by the same worker. If the maximum number of clients is set to one, then a given worker will not perform another call while processing, allowing other workers a chance to process it.
cert_file = /etc/swift/proxy.crt	to the ssl .crt. This should be enabled for testing purposes only.
key_file = /etc/swift/proxy.key	to the ssl .key. This should be enabled for testing purposes only.
expiring_objects_container_divisor = 86400	No help text available for this option.
expiring_objects_account_name = expiring_objects	No help text available for this option.
log_name = swift	Label used when logging
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_headers = false	No help text available for this option.
log_address = /dev/log	Location where syslog sends the logs to
trans_id_suffix =	No help text available for this option.
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_udp_host =	If not set, the UDB receiver for syslog is disabled.
log_udp_port = 514	Port value for UDB receiver, if enabled.
log_statsd_host = localhost	If not set, the StatsD feature is disabled.
log_statsd_port = 8125	Port value for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_sample_rate_factor = 1.0	Not recommended to set this to a value less than 1.0, if frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
cors_allow_origin =	is a list of hosts that are included with any CORS request by default and returned with the Access-Control-Allow-Origin header in addition to what the container has set. to call to setup custom log handlers. for eventlet the proxy server. For most cases, this should be `egg:swift#proxy`. request whenever it has to failover to a handoff node
client_timeout = 60	Timeout to read one chunk from a client external services
eventlet_debug = false	If true, turn on debug logging for eventlet

Table 8.34. Description of configuration options for [app:proxy-server] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#proxy	Entry point of paste.deploy in the server
set log_name = proxy-server	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_address = /dev/log	Location where syslog sends the logs to

Configuration option = Default value	Description
log_handoffs = true	No help text available for this option.
recheck_account_existence = 60	Cache timeout in seconds to send memcached for account existence
recheck_container_existence = 60	Cache timeout in seconds to send memcached for container existence
object_chunk_size = 8192	Chunk size to read from object servers
client_chunk_size = 8192	Chunk size to read from clients
node_timeout = 10	Request timeout to external services
recoverable_node_timeout = node_timeout	Request timeout to external services for requests that, on failure, can be recovered from. For example, object GET. from a client external services
conn_timeout = 0.5	Connection timeout to external services
post_quorum_timeout = 0.5	No help text available for this option.
error_suppression_interval = 60	Time in seconds that must elapse since the last error for a node to be considered no longer error limited
error_suppression_limit = 10	Error count to consider a node error limited
allow_account_management = false	Whether account PUTs and DELETEs are even callable
object_post_as_copy = true	Set object_post_as_copy = false to turn on fast posts where only the metadata changes are stored anew and the original data file is kept in place. This makes for quicker posts; but since the container metadata isn't updated in this mode, features like container sync won't be able to sync posts.
account_autocreate = false	If set to 'true' authorized accounts that do not yet exist within the Swift cluster will be automatically created.
max_containers_per_account = 0	If set to a positive value, trying to create a container when the account already has at least this maximum containers will result in a 403 Forbidden. Note: This is a soft limit, meaning a user might exceed the cap for recheck_account_existence before the 403s kick in.
max_containers_whitelist =	is a comma separated list of account names that ignore the max_containers_per_account cap.
deny_host_headers =	No help text available for this option.
auto_create_account_prefix = .	Prefix to use when automatically creating accounts
put_queue_depth = 10	No help text available for this option.
sorting_method = shuffle	No help text available for this option.
timing_expiry = 300	No help text available for this option.
max_large_object_get_time = 86400	No help text available for this option.
request_node_count = 2 * replicas	* replicas Set to the number of nodes to contact for a normal request. You can use '* replicas' at the end to have it use the number given times the number of replicas for the ring being used for the request. conf file for values will only be shown to the list of swift_owners. The exact default definition of a swift_owner is headers> up to the auth system in use, but usually indicates administrative responsibilities. paste.deploy to use for auth. To use tempauth set to: `egg:swift#tempauth` each request
read_affinity = r1z1=100, r1z2=200, r2=300	No help text available for this option.
read_affinity =	No help text available for this option.
write_affinity = r1, r2	No help text available for this option.
write_affinity =	No help text available for this option.
write_affinity_node_count = 2 * replicas	No help text available for this option.

Configuration option = Default value	Description
swift_owner_headers = x-container-read, x-container-write, x-container-sync-key, x-container-sync-to, x-account-meta-temp-url-key, x-account-meta-temp-url-key-2, x-account-access-control	the sample These are the headers whose conf file for values will only be shown to the list of swift_owners. The exact default definition of a swift_owner is headers> up to the auth system in use, but usually indicates administrative responsibilities. paste.deploy to use for auth. To use tempauth set to: `egg:swift#tempauth` each request

Table 8.35. Description of configuration options for [pipeline:main] in proxy-server.conf-sample

Configuration option = Default value	Description
pipeline = catch_errors gatekeeper healthcheck proxy-logging cache container_sync bulk tempurl slo dlo ratelimit tempauth container-quotas account-quotas proxy-logging proxy-server	No help text available for this option.

Table 8.36. Description of configuration options for [filter:account-quotas] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#account_quotas	Entry point of paste.deploy in the server

Table 8.37. Description of configuration options for [filter:authtoken] in proxy-server.conf-sample

Configuration option = Default value	Description
auth_host = keystonehost	No help text available for this option.
auth_port = 35357	No help text available for this option.
auth_protocol = http	No help text available for this option.
auth_uri = http://keystonehost:5000/	No help text available for this option.
admin_tenant_name = service	No help text available for this option.
admin_user = swift	No help text available for this option.
admin_password = password	No help text available for this option.
delay_auth_decision = 1	No help text available for this option.
cache = swift.cache	No help text available for this option.
include_service_catalog = False	No help text available for this option.

Table 8.38. Description of configuration options for [filter:cache] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#memcache	Entry point of paste.deploy in the server
set log_name = cache	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to
memcache_servers = 127.0.0.1:11211	Comma separated list of memcached servers ip:port services
memcache_serialization_support = 2	No help text available for this option.

Configuration option = Default value	Description
memcache_max_connections = 2	Max number of connections to each memcached server per worker services

Table 8.39. Description of configuration options for [filter:catch_errors] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#catch_errors	Entry point of paste.deploy in the server
set log_name = catch_errors	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to

Table 8.40. Description of configuration options for [filter:dlo] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#dlo	Entry point of paste.deploy in the server
rate_limit_after_segment = 10	Rate limit the download of large object segments after this segment is downloaded.
rate_limit_segments_per_sec = 1	Rate limit large object downloads at this rate. contact for a normal request. You can use '*' replicas' at the end to have it use the number given times the number of replicas for the ring being used for the request. paste.deploy to use for auth. To use tempauth set to: `egg:swift#tempauth` each request
max_get_time = 86400	No help text available for this option.

Table 8.41. Description of configuration options for [filter:gatekeeper] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#gatekeeper	Entry point of paste.deploy in the server
set log_name = gatekeeper	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to

Table 8.42. Description of configuration options for [filter:healthcheck] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#healthcheck	Entry point of paste.deploy in the server
disable_path =	No help text available for this option.

Table 8.43. Description of configuration options for [filter:keystoneauth] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#keystoneauth	Entry point of paste.deploy in the server
operator_roles = admin, swiftoperator	No help text available for this option.
reseller_admin_role = ResellerAdmin	No help text available for this option.

Table 8.44. Description of configuration options for [filter:list-endpoints] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#list_endpoints	Entry point of paste.deploy in the server
list_endpoints_path = /endpoints/	No help text available for this option.

Table 8.45. Description of configuration options for [filter:proxy-logging] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#proxy_logging	Entry point of paste.deploy in the server
access_log_name = swift	No help text available for this option.
access_log_facility = LOG_LOCAL0	No help text available for this option.
access_log_level = INFO	No help text available for this option.
access_log_address = /dev/log	No help text available for this option.
access_log_udp_host =	No help text available for this option.
access_log_udp_port = 514	No help text available for this option.
access_log_statsd_host = localhost	No help text available for this option.
access_log_statsd_port = 8125	No help text available for this option.
access_log_statsd_default_sample_rate = 1.0	No help text available for this option.
access_log_statsd_sample_rate_factor = 1.0	No help text available for this option.
access_log_statsd_metric_prefix =	No help text available for this option.
access_log_headers = false	No help text available for this option.
access_log_headers_only =	No help text available for this option.
logged with access_log_headers = True.	No help text available for this option.
reveal_sensitive_prefix = 8192	The X-Auth-Token is sensitive data. If revealed to an unauthorised person, they can now make requests against an account until the token expires. Set reveal_sensitive_prefix to the number of characters of the token that are logged. For example reveal_sensitive_prefix = 12 so only first 12 characters of the token are logged. Or, set to 0 to completely remove the token.
log_statsd_valid_http_methods = GET,HEAD,POST,PUT,DELETE,COPY,OPTIONS	No help text available for this option.

Table 8.46. Description of configuration options for [filter:tempauth] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#tempauth	Entry point of paste.deploy in the server
set log_name = tempauth	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility

Configuration option = Default value	Description
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to
reseller_prefix = AUTH	The naming scope for the auth service. Swift
auth_prefix = /auth/	The HTTP request path prefix for the auth service. Swift itself reserves anything beginning with the letter `v`.
token_life = 86400	The number of seconds a token is valid.
allow_overrides = true	No help text available for this option.
storage_url_scheme = default	Scheme to return with storage urls: http, https, or default (chooses based on what the server is running as) This can be useful with an SSL load balancer in front of a non-SSL server.
user_admin_admin = admin .admin .reseller_admin	No help text available for this option.
user_test_tester = testing .admin	No help text available for this option.
user_test2_tester2 = testing2 .admin	No help text available for this option.
user_test_tester3 = testing3	No help text available for this option.

Sample proxy server configuration file

```
[DEFAULT]
# bind_ip = 0.0.0.0
# bind_port = 80
# bind_timeout = 30
# backlog = 4096
# swift_dir = /etc/swift
# user = swift

# Enables exposing configuration settings via HTTP GET /info.
# expose_info = true

# Key to use for admin calls that are HMAC signed. Default is empty,
# which will disable admin calls to /info.
# admin_key = secret_admin_key
#
# Allows the ability to withhold sections from showing up in the public
# calls to /info. The following would cause the sections 'container_quotas'
# and 'tempurl' to not be listed. Default is empty, allowing all registered
# fetures to be listed via HTTP GET /info.
# disallowed_sections = container_quotas, tempurl

# Use an integer to override the number of pre-forked processes that will
# accept connections. Should default to the number of effective cpu
# cores in the system. It's worth noting that individual workers will
# use many eventlet co-routines to service multiple concurrent requests.
# workers = auto
#
# Maximum concurrent requests per worker
# max_clients = 1024
#
# Set the following two lines to enable SSL. This is for testing only.
# cert_file = /etc/swift/proxy.crt
# key_file = /etc/swift/proxy.key
#
# expiring_objects_container_divisor = 86400
# expiring_objects_account_name = expiring_objects
```

```
#
# You can specify default log routing here if you want:
# log_name = swift
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_headers = false
# log_address = /dev/log
#
# This optional suffix (default is empty) that would be appended to the swift
transaction
# id allows one to easily figure out from which cluster that X-Trans-Id
belongs to.
# This is very useful when one is managing more than one swift cluster.
# trans_id_suffix =
#
# comma separated list of functions to call to setup custom log handlers.
# functions get passed: conf, name, log_to_console, log_route, fmt, logger,
# adapted_logger
# log_custom_handlers =
#
# If set, log_udp_host will override log_address
# log_udp_host =
# log_udp_port = 514
#
# You can enable StatsD logging here:
# log_statsd_host = localhost
# log_statsd_port = 8125
# log_statsd_default_sample_rate = 1.0
# log_statsd_sample_rate_factor = 1.0
# log_statsd_metric_prefix =
#
# Use a comma separated list of full url (http://foo.bar:1234,https://foo.bar)
# cors_allow_origin =
#
# client_timeout = 60
# eventlet_debug = false

[pipeline:main]
pipeline = catch_errors gatekeeper healthcheck proxy-logging cache
container_sync bulk tempurl slo dlo ratelimit tempauth container-quotas
account-quotas proxy-logging proxy-server

[app:proxy-server]
use = egg:swift#proxy
# You can override the default log routing for this app here:
# set log_name = proxy-server
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_address = /dev/log
#
# log_handoffs = true
# recheck_account_existence = 60
# recheck_container_existence = 60
# object_chunk_size = 8192
# client_chunk_size = 8192
#
# How long the proxy server will wait on responses from the a/c/o servers.
# node_timeout = 10
#
# How long the proxy server will wait for an initial response and to read a
```

```
# chunk of data from the object servers while serving GET / HEAD requests.
# Timeouts from these requests can be recovered from so setting this to
# something lower than node_timeout would provide quicker error recovery
# while allowing for a longer timeout for non-recoverable requests (PUTs).
# Defaults to node_timeout, should be overridden if node_timeout is set to a
# high number to prevent client timeouts from firing before the proxy server
# has a chance to retry.
# recoverable_node_timeout = node_timeout
#
# conn_timeout = 0.5
#
# How long to wait for requests to finish after a quorum has been established.
# post_quorum_timeout = 0.5
#
# How long without an error before a node's error count is reset. This will
# also be how long before a node is reenabled after suppression is triggered.
# error_suppression_interval = 60
#
# How many errors can accumulate before a node is temporarily ignored.
# error_suppression_limit = 10
#
# If set to 'true' any authorized user may create and delete accounts; if
# 'false' no one, even authorized, can.
# allow_account_management = false
#
# Set object_post_as_copy = false to turn on fast posts where only the
# metadata
# changes are stored anew and the original data file is kept in place. This
# makes for quicker posts; but since the container metadata isn't updated in
# this mode, features like container sync won't be able to sync posts.
# object_post_as_copy = true
#
# If set to 'true' authorized accounts that do not yet exist within the Swift
# cluster will be automatically created.
# account_autocreate = false
#
# If set to a positive value, trying to create a container when the account
# already has at least this maximum containers will result in a 403 Forbidden.
# Note: This is a soft limit, meaning a user might exceed the cap for
# recheck_account_existence before the 403s kick in.
# max_containers_per_account = 0
#
# This is a comma separated list of account hashes that ignore the
# max_containers_per_account cap.
# max_containers_whitelist =
#
# Comma separated list of Host headers to which the proxy will deny requests.
# deny_host_headers =
#
# Prefix used when automatically creating accounts.
# auto_create_account_prefix = .
#
# Depth of the proxy put queue.
# put_queue_depth = 10
#
# Storage nodes can be chosen at random (shuffle), by using timing
# measurements (timing), or by using an explicit match (affinity).
# Using timing measurements may allow for lower overall latency, while
# using affinity allows for finer control. In both the timing and
# affinity cases, equally-sorting nodes are still randomly chosen to
```

```
# spread load.
# The valid values for sorting_method are "affinity", "shuffle", and "timing".
# sorting_method = shuffle
#
# If the "timing" sorting_method is used, the timings will only be valid for
# the number of seconds configured by timing_expiry.
# timing_expiry = 300
#
# The maximum time (seconds) that a large object connection is allowed to
# last.
# max_large_object_get_time = 86400
#
# Set to the number of nodes to contact for a normal request. You can use
# '* replicas' at the end to have it use the number given times the number of
# replicas for the ring being used for the request.
# request_node_count = 2 * replicas
#
# Which backend servers to prefer on reads. Format is r<N> for region
# N or r<N>z<M> for region N, zone M. The value after the equals is
# the priority; lower numbers are higher priority.
#
# Example: first read from region 1 zone 1, then region 1 zone 2, then
# anything in region 2, then everything else:
# read_affinity = r1z1=100, r1z2=200, r2=300
# Default is empty, meaning no preference.
# read_affinity =
#
# Which backend servers to prefer on writes. Format is r<N> for region
# N or r<N>z<M> for region N, zone M. If this is set, then when
# handling an object PUT request, some number (see setting
# write_affinity_node_count) of local backend servers will be tried
# before any nonlocal ones.
#
# Example: try to write to regions 1 and 2 before writing to any other
# nodes:
# write_affinity = r1, r2
# Default is empty, meaning no preference.
# write_affinity =
#
# The number of local (as governed by the write_affinity setting)
# nodes to attempt to contact first, before any non-local ones. You
# can use '* replicas' at the end to have it use the number given
# times the number of replicas for the ring being used for the
# request.
# write_affinity_node_count = 2 * replicas
#
# These are the headers whose values will only be shown to swift_owners. The
# exact definition of a swift_owner is up to the auth system in use, but
# usually indicates administrative responsibilities.
# swift_owner_headers = x-container-read, x-container-write, x-container-sync-
# key, x-container-sync-to, x-account-meta-temp-url-key, x-account-meta-temp-
# url-key-2, x-account-access-control

[filter:tempauth]
use = egg:swift#tempauth
# You can override the default log routing for this filter here:
# set log_name = tempauth
# set log_facility = LOG_LOCAL0
# set log_level = INFO
```



```
[filter:ratelimit]
use = egg:swift#ratelimit
# You can override the default log routing for this filter here:
# set log_name = ratelimit
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log
#
# clock_accuracy should represent how accurate the proxy servers' system
# clocks
# are with each other. 1000 means that all the proxies' clock are accurate to
# each other within 1 millisecond. No ratelimit should be higher than the
# clock accuracy.
# clock_accuracy = 1000
#
# max_sleep_time_seconds = 60
#
# log_sleep_time_seconds of 0 means disabled
# log_sleep_time_seconds = 0
#
# allows for slow rates (e.g. running up to 5 sec's behind) to catch up.
# rate_buffer_seconds = 5
#
# account_ratelimit of 0 means disabled
# account_ratelimit = 0

# these are comma separated lists of account names
# account_whitelist = a,b
# account_blacklist = c,d

# with container_limit_x = r
# for containers of size x limit write requests per second to r. The
# container
# rate will be linearly interpolated from the values given. With the values
# below, a container of size 5 will get a rate of 75.
# container_ratelimit_0 = 100
# container_ratelimit_10 = 50
# container_ratelimit_50 = 20

# Similarly to the above container-level write limits, the following will
# limit
# container GET (listing) requests.
# container_listing_ratelimit_0 = 100
# container_listing_ratelimit_10 = 50
# container_listing_ratelimit_50 = 20

[filter:domain_remap]
use = egg:swift#domain_remap
# You can override the default log routing for this filter here:
# set log_name = domain_remap
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log
#
# storage_domain = example.com
# path_root = v1
# reseller_prefixes = AUTH
```

```
[filter:catch_errors]
use = egg:swift#catch_errors
# You can override the default log routing for this filter here:
# set log_name = catch_errors
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log

[filter:cname_lookup]
# Note: this middleware requires python-dnspython
use = egg:swift#cname_lookup
# You can override the default log routing for this filter here:
# set log_name = cname_lookup
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log
#
# Specify the storage_domain that match your cloud, multiple domains
# can be specified separated by a comma
# storage_domain = example.com
#
# lookup_depth = 1

# Note: Put staticweb just after your auth filter(s) in the pipeline
[filter:staticweb]
use = egg:swift#staticweb

# Note: Put tempurl before dlo, slo and your auth filter(s) in the pipeline
[filter:tempurl]
use = egg:swift#tempurl
# The methods allowed with Temp URLs.
# methods = GET HEAD PUT
#
# The headers to remove from incoming requests. Simply a whitespace delimited
# list of header names and names can optionally end with '*' to indicate a
# prefix match. incoming_allow_headers is a list of exceptions to these
# removals.
# incoming_remove_headers = x-timestamp
#
# The headers allowed as exceptions to incoming_remove_headers. Simply a
# whitespace delimited list of header names and names can optionally end with
# '*' to indicate a prefix match.
# incoming_allow_headers =
#
# The headers to remove from outgoing responses. Simply a whitespace delimited
# list of header names and names can optionally end with '*' to indicate a
# prefix match. outgoing_allow_headers is a list of exceptions to these
# removals.
# outgoing_remove_headers = x-object-meta-*
#
# The headers allowed as exceptions to outgoing_remove_headers. Simply a
# whitespace delimited list of header names and names can optionally end with
# '*' to indicate a prefix match.
# outgoing_allow_headers = x-object-meta-public-*

# Note: Put formpost just before your auth filter(s) in the pipeline
[filter:formpost]
use = egg:swift#formpost
```



```
# Note: Put before both ratelimit and auth in the pipeline.
[filter:bulk]
use = egg:swift#bulk
# max_containers_per_extraction = 10000
# max_failed_extractions = 1000
# max_deletes_per_request = 10000
# max_failed_deletes = 1000

# In order to keep a connection active during a potentially long bulk request,
# Swift may return whitespace prepended to the actual response body. This
# whitespace will be yielded no more than every yield_frequency seconds.
# yield_frequency = 10

# Note: The following parameter is used during a bulk delete of objects and
# their container. This would frequently fail because it is very likely
# that all replicated objects have not been deleted by the time the middleware
# got a
# successful response. It can be configured the number of retries. And the
# number of seconds to wait between each retry will be 1.5**retry

# delete_container_retry_count = 0

# Note: Put after auth in the pipeline.
[filter:container-quotas]
use = egg:swift#container-quotas

# Note: Put before both ratelimit and auth in the pipeline.
[filter:slo]
use = egg:swift#slo
# max_manifest_segments = 1000
# max_manifest_size = 2097152
# min_segment_size = 1048576
# Start rate-limiting SLO segment serving after the Nth segment of a
# segmented object.
# rate_limit_after_segment = 10
#
# Once segment rate-limiting kicks in for an object, limit segments served
# to N per second. 0 means no rate-limiting.
# rate_limit_segments_per_sec = 0
#
# Time limit on GET requests (seconds)
# max_get_time = 86400

# Note: Put before both ratelimit and auth in the pipeline, but after
# gatekeeper, catch_errors, and proxy_logging (the first instance).
# If you don't put it in the pipeline, it will be inserted for you.
[filter:dlo]
use = egg:swift#dlo
# Start rate-limiting DLO segment serving after the Nth segment of a
# segmented object.
# rate_limit_after_segment = 10
#
# Once segment rate-limiting kicks in for an object, limit segments served
# to N per second. 0 means no rate-limiting.
# rate_limit_segments_per_sec = 1
#
# Time limit on GET requests (seconds)
# max_get_time = 86400
```

```
[filter:account-quotas]
use = egg:swift#account_quotas

[filter:gatekeeper]
use = egg:swift#gatekeeper
# You can override the default log routing for this filter here:
# set log_name = gatekeeper
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log

[filter:container_sync]
use = egg:swift#container_sync
# Set this to false if you want to disallow any full url values to be set for
# any new X-Container-Sync-To headers. This will keep any new full urls from
# coming in, but won't change any existing values already in the cluster.
# Updating those will have to be done manually, as knowing what the true realm
# endpoint should be cannot always be guessed.
# allow_full_urls = true
```

Proxy server memcache configuration

Find an example memcache configuration for the proxy server at `etc/memcache.conf-sample` in the source code repository.

The available configuration options are:

Table 8.47. Description of configuration options for [memcache] in `memcache.conf-sample`

Configuration option = Default value	Description
<code>memcache_servers = 127.0.0.1:11211</code>	Comma separated list of memcached servers ip:port services
<code>memcache_serialization_support = 2</code>	No help text available for this option.
<code>memcache_max_connections = 2</code>	Max number of connections to each memcached server per worker services

Rsyncd configuration

Find an example rsyncd configuration at `etc/rsyncd.conf-sample` in the source code repository.

The available configuration options are:

Table 8.48. Description of configuration options for [account] in `rsyncd.conf-sample`

Configuration option = Default value	Description
<code>max connections = 2</code>	No help text available for this option.
<code>path = /srv/node</code>	No help text available for this option.
<code>read only = false</code>	No help text available for this option.
<code>lock file = /var/lock/account.lock</code>	No help text available for this option.

Table 8.49. Description of configuration options for [container] in rsyncd.conf-sample

Configuration option = Default value	Description
max connections = 4	No help text available for this option.
path = /srv/node	No help text available for this option.
read only = false	No help text available for this option.
lock file = /var/lock/container.lock	No help text available for this option.

Table 8.50. Description of configuration options for [object] in rsyncd.conf-sample

Configuration option = Default value	Description
max connections = 8	No help text available for this option.
path = /srv/node	No help text available for this option.
read only = false	No help text available for this option.
lock file = /var/lock/object.lock	No help text available for this option.

Configure Object Storage features

Object Storage zones

In OpenStack Object Storage, data is placed across different tiers of failure domains. First, data is spread across regions, then zones, then servers, and finally across drives. Data is placed to get the highest failure domain isolation. If you deploy multiple regions, the Object Storage service places the data across the regions. Within a region, each replica of the data should be stored in unique zones, if possible. If there is only one zone, data should be placed on different servers. And if there is only one server, data should be placed on different drives.

Regions are widely separated installations with a high-latency or otherwise constrained network link between them. Zones are arbitrarily assigned, and it is up to the administrator of the Object Storage cluster to choose an isolation level and attempt to maintain the isolation level through appropriate zone assignment. For example, a zone may be defined as a rack with a single power source. Or a zone may be a DC room with a common utility provider. Servers are identified by a unique IP/port. Drives are locally attached storage volumes identified by mount point.

In small clusters (five nodes or fewer), everything is normally in a single zone. Larger Object Storage deployments may assign zone designations differently; for example, an entire cabinet or rack of servers may be designated as a single zone to maintain replica availability if the cabinet becomes unavailable (for example, due to failure of the top of rack switches or a dedicated circuit). In very large deployments, such as service provider level deployments, each zone might have an entirely autonomous switching and power infrastructure, so that even the loss of an electrical circuit or switching aggregator would result in the loss of a single replica at most.

Rackspace zone recommendations

For ease of maintenance on OpenStack Object Storage, Rackspace recommends that you set up at least five nodes. Each node is assigned its own zone (for a total of five zones),

which gives you host level redundancy. This enables you to take down a single zone for maintenance and still guarantee object availability in the event that another zone fails during your maintenance.

You could keep each server in its own cabinet to achieve cabinet level isolation, but you may wish to wait until your Object Storage service is better established before developing cabinet-level isolation. OpenStack Object Storage is flexible; if you later decide to change the isolation level, you can take down one zone at a time and move them to appropriate new homes.

RAID controller configuration

OpenStack Object Storage does not require RAID. In fact, most RAID configurations cause significant performance degradation. The main reason for using a RAID controller is the battery-backed cache. It is very important for data integrity reasons that when the operating system confirms a write has been committed that the write has actually been committed to a persistent location. Most disks lie about hardware commits by default, instead writing to a faster write cache for performance reasons. In most cases, that write cache exists only in non-persistent memory. In the case of a loss of power, this data may never actually get committed to disk, resulting in discrepancies that the underlying file system must handle.

OpenStack Object Storage works best on the XFS file system, and this document assumes that the hardware being used is configured appropriately to be mounted with the **nobarriers** option. For more information, refer to the XFS FAQ: http://xfs.org/index.php/XFS_FAQ

To get the most out of your hardware, it is essential that every disk used in OpenStack Object Storage is configured as a standalone, individual RAID 0 disk; in the case of 6 disks, you would have six RAID 0s or one JBOD. Some RAID controllers do not support JBOD or do not support battery backed cache with JBOD. To ensure the integrity of your data, you must ensure that the individual drive caches are disabled and the battery backed cache in your RAID card is configured and used. Failure to configure the controller properly in this case puts data at risk in the case of sudden loss of power.

You can also use hybrid drives or similar options for battery backed up cache configurations without a RAID controller.

Throttle resources through rate limits

Rate limiting in OpenStack Object Storage is implemented as a pluggable middleware that you configure on the proxy server. Rate limiting is performed on requests that result in database writes to the account and container SQLite databases. It uses memcached and is dependent on the proxy servers having highly synchronized time. The rate limits are limited by the accuracy of the proxy server clocks.

Configure rate limiting

All configuration is optional. If no account or container limits are provided, no rate limiting occurs. Available configuration options include:

Table 8.51. Description of configuration options for [filter:ratelimit] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#ratelimit	Entry point of paste.deploy in the server
set log_name = ratelimit	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to
clock_accuracy = 1000	Represents how accurate the proxy servers' system clocks are with each other. 1000 means that all the proxies' clock are accurate to each other within 1 millisecond. No ratelimit should be higher than the clock accuracy.
max_sleep_time_seconds = 60	App will immediately return a 498 response if the necessary sleep time ever exceeds the given max_sleep_time_seconds.
log_sleep_time_seconds = 0	To allow visibility into rate limiting set this value > 0 and all sleeps greater than the number will be logged.
rate_buffer_seconds = 5	Number of seconds the rate counter can drop and be allowed to catch up (at a faster than listed rate). A larger number will result in larger spikes in rate but better average accuracy.
account_ratelimit = 0	If set, will limit PUT and DELETE requests to /account_name/container_name. Number is in requests per second.
account_whitelist = a,b	Comma separated lists of account names that will not be rate limited.
account_blacklist = c,d	Comma separated lists of account names that will not be allowed. Returns a 497 response. r: for containers of size x, limit requests per second to r. Will limit PUT, DELETE, and POST requests to /a/c/o. container_listing_ratelimit_x = r: for containers of size x, limit listing requests per second to r. Will limit GET requests to /a/c.
with container_limit_x = r	No help text available for this option.
container_ratelimit_0 = 100	No help text available for this option.
container_ratelimit_10 = 50	No help text available for this option.
container_ratelimit_50 = 20	No help text available for this option.
container_listing_ratelimit_0 = 100	No help text available for this option.
container_listing_ratelimit_10 = 50	No help text available for this option.
container_listing_ratelimit_50 = 20	No help text available for this option.

The container rate limits are linearly interpolated from the values given. A sample container rate limiting could be:

```
container_ratelimit_100 = 100
```

```
container_ratelimit_200 = 50
```

```
container_ratelimit_500 = 20
```

This would result in:

Table 8.52. Values for Rate Limiting with Sample Configuration Settings

Container Size	Rate Limit
0-99	No limiting
100	100
150	75
500	20
1000	20

Health check

Provides an easy way to monitor whether the Object Storage proxy server is alive. If you access the proxy with the path `/healthcheck`, it responds with OK in the response body, which monitoring tools can use.

Table 8.53. Description of configuration options for [filter:healthcheck] in account-server.conf-sample

Configuration option = Default value	Description
<code>use = egg:swift#healthcheck</code>	Entry point of paste.deploy in the server
<code>disable_path =</code>	No help text available for this option.

Domain remap

Middleware that translates container and account parts of a domain to path parameters that the proxy server understands.

Table 8.54. Description of configuration options for [filter:domain_remap] in proxy-server.conf-sample

Configuration option = Default value	Description
<code>use = egg:swift#domain_remap</code>	Entry point of paste.deploy in the server
<code>set log_name = domain_remap</code>	Label to use when logging
<code>set log_facility = LOG_LOCAL0</code>	Syslog log facility
<code>set log_level = INFO</code>	Log level
<code>set log_headers = false</code>	If True, log headers in each request
<code>set log_address = /dev/log</code>	Location where syslog sends the logs to
<code>storage_domain = example.com</code>	Domain that matches your cloud. Multiple domains can be specified using a comma-separated list.
<code>path_root = v1</code>	Root path
<code>reseller_prefixes = AUTH</code>	Reseller prefix

CNAME lookup

Middleware that translates an unknown domain in the host header to something that ends with the configured `storage_domain` by looking up the given domain's CNAME record in DNS.

**Table 8.55. Description of configuration options for
[filter:cname_lookup] in proxy-server.conf-sample**

Configuration option = Default value	Description
use = egg:swift#cname_lookup	Entry point of paste.deploy in the server
set log_name = cname_lookup	Label to use when logging
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_headers = false	If True, log headers in each request
set log_address = /dev/log	Location where syslog sends the logs to
storage_domain = example.com	Domain that matches your cloud. Multiple domains can be specified using a comma-separated list.
lookup_depth = 1	Because CNAMEs can be recursive, specifies the number of levels through which to search.

Temporary URL

Allows the creation of URLs to provide temporary access to objects. For example, a website may wish to provide a link to download a large object in OpenStack Object Storage, but the Object Storage account has no public access. The website can generate a URL that provides GET access for a limited time to the resource. When the web browser user clicks on the link, the browser downloads the object directly from Object Storage, eliminating the need for the website to act as a proxy for the request. If the user shares the link with all his friends, or accidentally posts it on a forum, the direct access is limited to the expiration time set when the website created the link.

A temporary URL is the typical URL associated with an object, with two additional query parameters:

`temp_url_sig` A cryptographic signature
`temp_url_expires` An expiration date, in Unix time

An example of a temporary URL:

```
https://swift-cluster.example.com/v1/AUTH_a422b2-91f3-2f46-74b7-
d7c9e8958f5d30/container/object?
temp_url_sig=da39a3ee5e6b4b0d3255bfef95601890afd80709&
temp_url_expires=1323479485
```

To create temporary URLs, first set the `X-Account-Meta-Temp-URL-Key` header on your Object Storage account to an arbitrary string. This string serves as a secret key. For example, to set a key of `b3968d0207b54ece87cccc06515a89d4` using the `swift` command-line tool:

```
$ swift post -m "Temp-URL-Key:b3968d0207b54ece87cccc06515a89d4"
```

Next, generate an HMAC-SHA1 (RFC 2104) signature to specify:

- Which HTTP method to allow (typically GET or PUT)
- The expiry date as a Unix timestamp
- The full path to the object

- The secret key set as the X-Account-Meta-Temp-URL-Key

Here is code generating the signature for a GET for 24 hours on `/v1/AUTH_account/container/object`:

```
import hmac
from hashlib import sha1
from time import time
method = 'GET'
duration_in_seconds = 60*60*24
expires = int(time() + duration_in_seconds)
path = '/v1/AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30/container/object'
key = 'mykey'
hmac_body = '%s\n%s\n%s' % (method, expires, path)
sig = hmac.new(key, hmac_body, sha1).hexdigest()
s = 'https://{host}/{path}?temp_url_sig={sig}&temp_url_expires={expires}'
url = s.format(host='swift-cluster.example.com', path=path, sig=sig, expires=
expires)
```

Any alteration of the resource path or query arguments results in a 401 Unauthorized error. Similarly, a PUT where GET was the allowed method returns a 401. HEAD is allowed if GET or PUT is allowed. Using this in combination with browser form post translation middleware could also allow direct-from-browser uploads to specific locations in Object Storage.



Note

Changing the X-Account-Meta-Temp-URL-Key invalidates any previously generated temporary URLs within 60 seconds (the memcache time for the key). Object Storage supports up to two keys, specified by X-Account-Meta-Temp-URL-Key and X-Account-Meta-Temp-URL-Key-2. Signatures are checked against both keys, if present. This is to allow for key rotation without invalidating all existing temporary URLs.

Object Storage includes a script called **swift-temp-url** that generates the query parameters automatically:

```
$ bin/swift-temp-url GET 3600 /v1/AUTH_account/container/object mykey
/v1/AUTH_account/container/object?
temp_url_sig=5c4cc8886f36a9d0919d708ade98bf0cc71c9e91&
temp_url_expires=1374497657
```

Because this command only returns the path, you must prefix the Object Storage host name (for example, `https://swift-cluster.example.com`).

With GET Temporary URLs, a Content-Disposition header is set on the response so that browsers interpret this as a file attachment to be saved. The file name chosen is based on the object name, but you can override this with a `filename` query parameter. The following example specifies a filename of `My Test File.pdf`:

```
https://swift-cluster.example.com/v1/AUTH_a422b2-91f3-2f46-74b7-
d7c9e8958f5d30/container/object?
temp_url_sig=da39a3ee5e6b4b0d3255bfef95601890afd80709&
temp_url_expires=1323479485&
filename=My+Test+File.pdf
```

To enable Temporary URL functionality, edit `/etc/swift/proxy-server.conf` to add `tempurl` to the pipeline variable defined in the `[pipeline:main]` section. The

`tempurl` entry should appear immediately before the authentication filters in the pipeline, such as `authtoken`, `tempauth` or `keystoneauth`. For example:

```
[pipeline:main]
pipeline = pipeline = healthcheck cache tempurl authtoken keystoneauth proxy-
server
```

Table 8.56. Description of configuration options for `[filter:tempurl]` in `proxy-server.conf-sample`

Configuration option = Default value	Description
<code>use = egg:swift#tempurl</code>	Entry point of paste.deploy in the server
<code>methods = GET HEAD PUT</code>	HTTP methods allowed with Temporary URLs
<code>incoming_remove_headers = x-timestamp</code>	Headers to remove from incoming requests. Simply a whitespace delimited list of header names and names can optionally end with '*' to indicate a prefix match.
<code>incoming_allow_headers =</code>	Headers allowed as exceptions to <code>incoming_remove_headers</code> . Simply a whitespace delimited list of header names and names can optionally end with '*' to indicate a prefix match.
<code>outgoing_remove_headers = x-object-meta-*</code>	Headers to remove from outgoing responses. Simply a whitespace delimited list of header names and names can optionally end with '*' to indicate a prefix match.
<code>outgoing_allow_headers = x-object-meta-public-*</code>	Headers allowed as exceptions to <code>outgoing_remove_headers</code> . Simply a whitespace delimited list of header names and names can optionally end with '*' to indicate a prefix match.

Name check filter

Name Check is a filter that disallows any paths that contain defined forbidden characters or that exceed a defined length.

Table 8.57. Description of configuration options for `[filter:name_check]` in `proxy-server.conf-sample`

Configuration option = Default value	Description
<code>use = egg:swift#name_check</code>	Entry point of paste.deploy in the server
<code>forbidden_chars = "" <></code>	Characters that are not allowed in a name
<code>maximum_length = 255</code>	Maximum length of a name
<code>forbidden_regexp = /\./ /\.\./ /\.\$ /\.\.\$</code>	Substrings to forbid, using regular expression syntax

Constraints

To change the OpenStack Object Storage internal limits, update the values in the `swift-constraints` section in the `swift.conf` file. Use caution when you update these values because they affect the performance in the entire cluster.

Table 8.58. Description of configuration options for `[swift-constraints]` in `swift.conf-sample`

Configuration option = Default value	Description
<code>max_file_size = 5368709122</code>	The largest normal object that can be saved in the cluster. This is also the limit on the size of each segment of a large

Configuration option = Default value	Description
	object when using the large object manifest support. This value is set in bytes. Setting it to lower than 1MiB will cause some tests to fail. It is STRONGLY recommended to leave this value at the default (5 * 2**30 + 2).
max_meta_name_length = 128	The maximum number of bytes in the utf8 encoding of the name portion of a metadata header.
max_meta_value_length = 256	The max number of bytes in the utf8 encoding of a metadata value.
max_meta_count = 90	The maximum number of metadata keys that can be stored on a single account, container, or object.
max_meta_overall_size = 4096	The maximum number of bytes in the utf8 encoding of the metadata (keys + values).
max_header_size = 8192	The maximum number of bytes in the utf8 encoding of each header.
max_object_name_length = 1024	The maximum number of bytes in the utf8 encoding of an object name.
container_listing_limit = 10000	The default (and maximum) number of items returned for a container listing request.
account_listing_limit = 10000	The default (and maximum) number of items returned for an account listing request.
max_account_name_length = 256	The maximum number of bytes in the utf8 encoding of an account name.
max_container_name_length = 256	The maximum number of bytes in the utf8 encoding of a container name.

Cluster health

Use the **swift-dispersion-report** tool to measure overall cluster health. This tool checks if a set of deliberately distributed containers and objects are currently in their proper places within the cluster. For instance, a common deployment has three replicas of each object. The health of that object can be measured by checking if each replica is in its proper place. If only 2 of the 3 is in place the object's health can be said to be at 66.66%, where 100% would be perfect. A single object's health, especially an older object, usually reflects the health of that entire partition the object is in. If you make enough objects on a distinct percentage of the partitions in the cluster, you get a good estimate of the overall cluster health. In practice, about 1% partition coverage seems to balance well between accuracy and the amount of time it takes to gather results. The first thing that needs to be done to provide this health value is create a new account solely for this usage. Next, you need to place the containers and objects throughout the system so that they are on distinct partitions. The **swift-dispersion-populate** tool does this by making up random container and object names until they fall on distinct partitions. Last, and repeatedly for the life of the cluster, you must run the **swift-dispersion-report** tool to check the health of each of these containers and objects. These tools need direct access to the entire cluster and to the ring files (installing them on a proxy server suffices). The **swift-dispersion-populate** and **swift-dispersion-report** commands both use the same configuration file, `/etc/swift/dispersion.conf`. Example `dispersion.conf` file:

```
[dispersion]
auth_url = http://localhost:8080/auth/v1.0
auth_user = test:tester
auth_key = testing
```

There are also configuration options for specifying the dispersion coverage, which defaults to 1%, retries, concurrency, and so on. However, the defaults are usually fine. Once the configuration is in place, run **swift-dispersion-populate** to populate the containers and objects throughout the cluster. Now that those containers and objects are in place, you can run **swift-dispersion-report** to get a dispersion report, or the overall health of the cluster. Here is an example of a cluster in perfect health:

```
$ swift-dispersion-report
Queried 2621 containers for dispersion reporting, 19s, 0 retries
100.00% of container copies found (7863 of 7863)
Sample represents 1.00% of the container partition space

Queried 2619 objects for dispersion reporting, 7s, 0 retries
100.00% of object copies found (7857 of 7857)
Sample represents 1.00% of the object partition space
```

Now, deliberately double the weight of a device in the object ring (with replication turned off) and re-run the dispersion report to show what impact that has:

```
$ swift-ring-builder object.builder set_weight d0 200
$ swift-ring-builder object.builder rebalance
...
$ swift-dispersion-report
Queried 2621 containers for dispersion reporting, 8s, 0 retries
100.00% of container copies found (7863 of 7863)
Sample represents 1.00% of the container partition space

Queried 2619 objects for dispersion reporting, 7s, 0 retries
There were 1763 partitions missing one copy.
77.56% of object copies found (6094 of 7857)
Sample represents 1.00% of the object partition space
```

You can see the health of the objects in the cluster has gone down significantly. Of course, this test environment has just four devices, in a production environment with many devices the impact of one device change is much less. Next, run the replicators to get everything put back into place and then rerun the dispersion report:

```
... start object replicators and monitor logs until they're caught up ...
$ swift-dispersion-report
Queried 2621 containers for dispersion reporting, 17s, 0 retries
100.00% of container copies found (7863 of 7863)
Sample represents 1.00% of the container partition space

Queried 2619 objects for dispersion reporting, 7s, 0 retries
100.00% of object copies found (7857 of 7857)
Sample represents 1.00% of the object partition space
```

Alternatively, the dispersion report can also be output in json format. This allows it to be more easily consumed by third-party utilities:

```
$ swift-dispersion-report -j
{"object": {"retries": 0, "missing_two": 0, "copies_found": 7863,
"missing_one": 0,
"copies_expected": 7863, "pct_found": 100.0, "overlapping": 0, "missing_all":
0}, "container":
{"retries": 0, "missing_two": 0, "copies_found": 12534, "missing_one": 0,
"copies_expected":
```

```
12534, "pct_found": 100.0, "overlapping": 15, "missing_all": 0}}
```

Table 8.59. Description of configuration options for [dispersion] in dispersion.conf-sample

Configuration option = Default value	Description
auth_url = http://localhost:8080/auth/v1.0	Endpoint for auth server, such as keystone
auth_user = test:tester	Default user for dispersion in this context
auth_key = testing	No help text available for this option.
auth_url = http://localhost:5000/v2.0/	Endpoint for auth server, such as keystone
auth_user = tenant:user	Default user for dispersion in this context
auth_key = password	No help text available for this option.
auth_version = 2.0	Indicates which version of auth
endpoint_type = publicURL	Indicates whether endpoint for auth is public or internal
keystone_api_insecure = no	No help text available for this option.
swift_dir = /etc/swift	Swift configuration directory
dispersion_coverage = 1.0	No help text available for this option.
retries = 5	No help text available for this option.
concurrency = 25	Number of replication workers to spawn
container_populate = yes	No help text available for this option.
object_populate = yes	No help text available for this option.
container_report = yes	No help text available for this option.
object_report = yes	No help text available for this option.
dump_json = no	No help text available for this option.

Static Large Object (SLO) support

This feature is very similar to Dynamic Large Object (DLO) support in that it enables the user to upload many objects concurrently and afterwards download them as a single object. It is different in that it does not rely on eventually consistent container listings to do so. Instead, a user-defined manifest of the object segments is used.

Table 8.60. Description of configuration options for [filter:slo] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#slo	Entry point of paste.deploy in the server
max_manifest_segments = 1000	No help text available for this option.
max_manifest_size = 2097152	No help text available for this option.
min_segment_size = 1048576	No help text available for this option.
rate_limit_after_segment = 10	Rate limit the download of large object segments after this segment is downloaded.
rate_limit_segments_per_sec = 0	Rate limit large object downloads at this rate. contact for a normal request. You can use '*' replicas' at the end to have it use the number given times the number of replicas for the ring being used for the request. paste.deploy to use for auth. To use tempauth set to: `egg:swift#tempauth` each request
max_get_time = 86400	No help text available for this option.

Container quotas

The `container_quotas` middleware implements simple quotas that can be imposed on Object Storage containers by a user with the ability to set container metadata, most likely the account administrator. This can be useful for limiting the scope of containers that are delegated to non-admin users, exposed to formpost uploads, or just as a self-imposed sanity check.

Any object PUT operations that exceed these quotas return a 413 response (request entity too large) with a descriptive body.

Quotas are subject to several limitations: eventual consistency, the timeliness of the cached `container_info` (60 second ttl by default), and it is unable to reject chunked transfer uploads that exceed the quota (though once the quota is exceeded, new chunked transfers are refused).

Set quotas by adding meta values to the container. These values are validated when you set them:

- `X-Container-Meta-Quota-Bytes`: Maximum size of the container, in bytes.
- `X-Container-Meta-Quota-Count`: Maximum object count of the container.

Table 8.61. Description of configuration options for `[filter:container-quotas]` in `proxy-server.conf-sample`

Configuration option = Default value	Description
<code>use = egg:swift#container_quotas</code>	Entry point of <code>paste.deploy</code> in the server

Account quotas

The `x-account-meta-quota-bytes` metadata entry must be requests (PUT, POST) if a given account quota (in bytes) is exceeded while DELETE requests are still allowed.

The `x-account-meta-quota-bytes` metadata entry must be set to store and enable the quota. Write requests to this metadata entry are only permitted for resellers. There is no account quota limitation on a reseller account even if `x-account-meta-quota-bytes` is set.

Any object PUT operations that exceed the quota return a 413 response (request entity too large) with a descriptive body.

The following command uses an admin account that own the Reseller role to set a quota on the test account:

```
$ swift -A http://127.0.0.1:8080/auth/v1.0 -U admin:admin -K admin \  
--os-storage-url=http://127.0.0.1:8080/v1/AUTH_test post -m quota-bytes:10000
```

Here is the stat listing of an account where quota has been set:

```
$ swift -A http://127.0.0.1:8080/auth/v1.0 -U test:tester -K testing stat  
Account: AUTH_test  
Containers: 0  
Objects: 0
```



```
Bytes: 0
Meta Quota-Bytes: 10000
X-Timestamp: 1374075958.37454
X-Trans-Id: tx602634cf478546a39b1be-0051e6bc7a
```

This command removes the account quota:

```
$ swift -A http://127.0.0.1:8080/auth/v1.0 -U admin:admin -K admin --os-
storage-url=http://127.0.0.1:8080/v1/AUTH_test post -m quota-bytes:
```

Bulk delete

Use `bulk-delete` to delete multiple files from an account with a single request. Responds to DELETE requests with a header 'X-Bulk-Delete: true_value'. The body of the DELETE request is a new line-separated list of files to delete. The files listed must be URL encoded and in the form:

```
/container_name/obj_name
```

If all files are successfully deleted (or did not exist), the operation returns `HTTPOk`. If any files failed to delete, the operation returns `HTTPBadGateway`. In both cases, the response body is a JSON dictionary that shows the number of files that were successfully deleted or not found. The files that failed are listed.

Table 8.62. Description of configuration options for [filter:bulk] in `proxy-server.conf-sample`

Configuration option = Default value	Description
<code>use = egg:swift#bulk</code>	Entry point of paste.deploy in the server
<code>max_containers_per_extraction = 10000</code>	No help text available for this option.
<code>max_failed_extractions = 1000</code>	No help text available for this option.
<code>max_deletes_per_request = 10000</code>	No help text available for this option.
<code>max_failed_deletes = 1000</code>	No help text available for this option.
<code>yield_frequency = 10</code>	No help text available for this option.
<code>delete_container_retry_count = 0</code>	No help text available for this option.

Configure Object Storage with the S3 API

The Swift3 middleware emulates the S3 REST API on top of Object Storage.

The following operations are currently supported:

- GET Service
- DELETE Bucket
- GET Bucket (List Objects)
- PUT Bucket
- DELETE Object

Drive audit

The `swift-drive-audit` configuration items reference a script that can be run by using `cron` to watch for bad drives. If errors are detected, it unmounts the bad drive, so that OpenStack Object Storage can work around it. It takes the following options:

Table 8.63. Description of configuration options for `[drive-audit]` in `drive-audit.conf-sample`

Configuration option = Default value	Description
<code>device_dir = /srv/node</code>	Directory devices are mounted under
<code>log_facility = LOG_LOCAL0</code>	Syslog log facility
<code>log_level = INFO</code>	Logging level
<code>log_address = /dev/log</code>	Location where syslog sends the logs to
<code>minutes = 60</code>	Number of minutes to look back in <code>`/var/log/kern.log`</code>
<code>error_limit = 1</code>	Number of errors to find before a device is unmounted
<code>log_file_pattern = /var/log/kern*</code>	Location of the log file with globbing pattern to check against device errors locate device blocks with errors in the log file
<code>regex_pattern_1 = \berror\b.*\b(dm-[0-9]{1,2}\d?)\b</code>	No help text available for this option.

Form post

Middleware that provides the ability to upload objects to a cluster using an HTML form POST. The format of the form is:

```
<![CDATA[
<form action="<swift-url>" method="POST"
  enctype="multipart/form-data">
  <input type="hidden" name="redirect-url" value="<redirect-url>" />
  <input type="hidden" name="max_file_size" value="<bytes>" />
  <input type="hidden" name="max_file_count" value="<count>" />
  <input type="hidden" name="expires" value="<unix-timestamp>" />
  <input type="hidden" name="signature" value="<hmac>" />
  <input type="file" name="file1" /><br />
  <input type="submit" />
</form>]]>
```

The `swift-url` is the URL to the Object Storage destination, such as: `https://swift-cluster.example.com/v1/AUTH_account/container/object_prefix` The name of each file uploaded is appended to the specified `swift-url`. So, you can upload directly to the root of container with a URL like: `https://swift-cluster.example.com/v1/AUTH_account/container/` Optionally, you can include an object prefix to better separate different users' uploads, such as: `https://swift-cluster.example.com/v1/AUTH_account/container/object_prefix`



Note

The form method must be `POST` and the `enctype` must be set as `multipart/form-data`.

The `redirect` attribute is the URL to redirect the browser to after the upload completes. The URL has status and message query parameters added to it, indicating the HTTP status code

for the upload (2xx is success) and a possible message for further information if there was an error (such as "max_file_size exceeded").

The `max_file_size` attribute must be included and indicates the largest single file upload that can be done, in bytes.

The `max_file_count` attribute must be included and indicates the maximum number of files that can be uploaded with the form. Include additional `<![CDATA[<input type="file" name="filexx"/>]]>` attributes if desired.

The `expires` attribute is the Unix timestamp before which the form must be submitted before it is invalidated.

The `signature` attribute is the HMAC-SHA1 signature of the form. This sample Python code shows how to compute the signature:

```
import hmac
from hashlib import sha1
from time import time
path = '/v1/account/container/object_prefix'
redirect = 'https://myserver.com/some-page'
max_file_size = 104857600
max_file_count = 10
expires = int(time() + 600)
key = 'mykey'
hmac_body = '%s\n%s\n%s\n%s\n%s' % (path, redirect,
    max_file_size, max_file_count, expires)
signature = hmac.new(key, hmac_body, sha1).hexdigest()
```

The key is the value of the `X-Account-Meta-Temp-URL-Key` header on the account.

Be certain to use the full path, from the `/v1/` onward.

The command-line tool `swift-form-signature` may be used (mostly just when testing) to compute `expires` and `signature`.

The file attributes must appear after the other attributes to be processed correctly. If attributes come after the file, they are not sent with the sub-request because on the server side, all attributes in the file cannot be parsed unless the whole file is read into memory and the server does not have enough memory to service these requests. So, attributes that follow the file are ignored.

Table 8.64. Description of configuration options for `[filter:formpost]` in `proxy-server.conf-sample`

Configuration option = Default value	Description
<code>use = egg:swift#formpost</code>	Entry point of <code>paste.deploy</code> in the server

Static web sites

When configured, this middleware serves container data as a static web site with index file and error file resolution and optional file listings. This mode is normally only active for anonymous requests.

Table 8.65. Description of configuration options for [filter:staticweb] in proxy-server.conf-sample

Configuration option = Default value	Description
use = egg:swift#staticweb	Entry point of paste.deploy in the server

Cross-origin resource sharing

Cross-Origin Resource Sharing (CORS) is a mechanism to allow code running in a browser (JavaScript for example) to make requests to a domain other than the one from where it originated. OpenStack Object Storage supports CORS requests to containers and objects within the containers using metadata held on the container.

In addition to the metadata on containers, you can use the `cors_allow_origin` option in the `proxy-server.conf` file to set a list of hosts that are included with any CORS request by default.

Endpoint listing middleware

The endpoint listing middleware enables third-party services that use data locality information to integrate with OpenStack Object Storage. This middleware reduces network overhead and is designed for third-party services that run inside the firewall. Deploy this middleware on a proxy server because usage of this middleware is not authenticated.

Format requests for endpoints, as follows:

```
/endpoints/{account}/{container}/{object}
/endpoints/{account}/{container}
/endpoints/{account}
```

Use the `list_endpoints_path` configuration option in the `proxy_server.conf` file to customize the `/endpoints/` path.

Responses are JSON-encoded lists of endpoints, as follows:

```
http://{server}:{port}/{dev}/{part}/{acc}/{cont}/{obj}
http://{server}:{port}/{dev}/{part}/{acc}/{cont}
http://{server}:{port}/{dev}/{part}/{acc}
```

An example response is:

```
http://10.1.1.1:6000/sda1/2/a/c2/o1
http://10.1.1.1:6000/sda1/2/a/c2
http://10.1.1.1:6000/sda1/2/a
```

9. Orchestration

Table of Contents

Configure APIs	487
Configure Clients	490
Configure the RPC messaging system	493
New, updated and deprecated options for heat	496

The Orchestration service is designed to manage the lifecycle of infrastructure and applications within OpenStack clouds. Its various agents and services are configured in the `/etc/heat/heat.conf` file.

To install Orchestration, see the *OpenStack Installation Guide* for your distribution (docs.openstack.org).

The following tables provide a comprehensive list of the Orchestration configuration options.

Table 9.1. Description of configuration options for `auth_token`

Configuration option = Default value	Description
[DEFAULT]	
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) Used to control the use and type of token binding. Can be set to: "disabled" to not check token

Configuration option = Default value	Description
	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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. 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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.

Table 9.2. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
deferred_auth_method = password	(StrOpt) Select deferred auth method, stored password or trusts.
environment_dir = /etc/heat/environment.d	(StrOpt) The directory to search for environment files.
event_purge_batch_size = 10	(IntOpt) Controls how many events will be pruned whenever a stack's events exceed max_events_per_stack. Set this lower to keep more events at the expense of more frequent purges.
instance_driver = heat.engine.nova	(StrOpt) Driver to use for controlling instances.
instance_user = ec2-user	(StrOpt) The default user for new instances. This option is deprecated and will be removed in the Juno release. If it's empty, Heat will use the default user set up with

Configuration option = Default value	Description
	your cloud image (for OS::Nova::Server) or 'ec2-user' (for AWS::EC2::Instance).
keystone_backend = heat.common.heat_keystoneclient.KeystoneClientV3	(StrOpt) Fully qualified class name to use as a keystone backend.
periodic_interval = 60	(IntOpt) Seconds between running periodic tasks.
plugin_dirs = /usr/lib64/heat, /usr/lib/heat	(ListOpt) List of directories to search for plug-ins.
[revision]	
heat_revision = unknown	(StrOpt) Heat build revision. If you would prefer to manage your build revision separately, you can move this section to a different file and add it as another config option.

Table 9.3. Description of configuration options for crypt

Configuration option = Default value	Description
[DEFAULT]	
auth_encryption_key = notgood but just long enough i think	(StrOpt) Encryption key used for authentication info in database.

Table 9.4. Description of configuration options for database

Configuration option = Default value	Description
[DEFAULT]	
db_backend = sqlalchemy	(StrOpt) The backend to use for db.
sqlite_db = heat.sqlite	(StrOpt) the filename to use with sqlite
sqlite_synchronous = True	(BoolOpt) If true, use synchronous mode for sqlite
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
connection = sqlite:///usr/lib/python/site-packages/heat/openstack/common/db/\$sqlite_db	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
idle_timeout = 3600	(IntOpt) timeout before idle sql connections are reaped
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
retry_interval = 10	(IntOpt) interval between retries of opening a sql connection
slave_connection =	(StrOpt) The SQLAlchemy connection string used to connect to the slave database

Table 9.5. Description of configuration options for debug

Configuration option = Default value	Description
[DEFAULT]	
backdoor_port = None	(StrOpt) 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 service's log file.
debug = False	(BoolOpt) Print debugging output (set logging level to DEBUG instead of default WARNING level).
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks
fatal_deprecations = False	(BoolOpt) make deprecations fatal
lock_path = None	(StrOpt) Directory to use for lock files.

Table 9.6. Description of configuration options for loadbalancer

Configuration option = Default value	Description
[DEFAULT]	
loadbalancer_template = None	(StrOpt) Custom template for the built-in loadbalancer nested stack.

Table 9.7. Description of configuration options for logging

Configuration option = Default value	Description
[DEFAULT]	
default_log_levels = amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN	(ListOpt) list of logger=LEVEL pairs
instance_format = "[instance: %(uuid)s] "	(StrOpt) If an instance is passed with the log message, format it like this
instance_uuid_format = "[instance: %(uuid)s] "	(StrOpt) If an instance UUID is passed with the log message, format it like this
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
log_date_format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %(asctime)s in log records. Default: %(default)s
log_dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths
log_file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
log_format = None	(StrOpt) DEPRECATED. A logging.Formatter log message format string which may use any of the available logging.LogRecord attributes. This option is deprecated. Please use logging_context_format_string and logging_default_format_string instead.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s %(instance)s %(message)s	(StrOpt) format string to use for log messages with context

Configuration option = Default value	Description
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	(StrOpt) data to append to log format when level is DEBUG
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) prefix each line of exception output with this format
syslog_log_facility = LOG_USER	(StrOpt) syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 9.8. Description of configuration options for quota

Configuration option = Default value	Description
[DEFAULT]	
max_events_per_stack = 1000	(IntOpt) Maximum events that will be available per stack. Older events will be deleted when this is reached. Set to 0 for unlimited events per stack.
max_nested_stack_depth = 3	(IntOpt) Maximum depth allowed when using nested stacks.
max_resources_per_stack = 1000	(IntOpt) Maximum resources allowed per top-level stack.
max_stacks_per_tenant = 100	(IntOpt) Maximum number of stacks any one tenant may have active at one time.
max_template_size = 524288	(IntOpt) Maximum raw byte size of any template.

Table 9.9. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	
host = oslo	(StrOpt) Name of the engine node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address.
[matchmaker_redis]	
host = 127.0.0.1	(StrOpt) Host to locate redis
password = None	(StrOpt) Password for Redis server. (optional)
port = 6379	(IntOpt) Use this port to connect to redis host.

Configure APIs

The following options allow configuration of the APIs that Orchestration supports. Currently this includes compatibility APIs for CloudFormation and CloudWatch and a native API.

Table 9.10. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
heat_metadata_server_url =	(StrOpt) URL of the Heat metadata server.
heat_stack_user_role = heat_stack_user	(StrOpt) Keystone role for heat template-defined users.

Configuration option = Default value	Description
heat_waitcondition_server_url =	(StrOpt) URL of the Heat waitcondition server.
heat_watch_server_url =	(StrOpt) URL of the Heat CloudWatch server.
max_json_body_size = 1048576	(IntOpt) Maximum raw byte size of JSON request body. Should be larger than max_template_size.
policy_default_rule = default	(StrOpt) Rule enforced when requested rule is not found
policy_file = policy.json	(StrOpt) JSON file containing policy
secure_proxy_ssl_header = X-Forwarded-Proto	(StrOpt) The HTTP Header that will be used to determine which the original request protocol scheme was, even if it was removed by an SSL terminator proxy.
stack_action_timeout = 3600	(IntOpt) Timeout in seconds for stack action (ie. create or update).
stack_domain_admin = None	(StrOpt) Keystone username, a user with roles sufficient to manage users and projects in the stack_user_domain.
stack_domain_admin_password = None	(StrOpt) Keystone password for stack_domain_admin user.
stack_user_domain = None	(StrOpt) Keystone domain ID which contains heat template-defined users.
trusts_delegated_roles = heat_stack_owner	(ListOpt) Subset of trustor roles to be delegated to heat.
[auth_password]	
allowed_auth_uris =	(ListOpt) Allowed keystone endpoints for auth_uri when multi_cloud is enabled. At least one endpoint needs to be specified.
multi_cloud = False	(BoolOpt) Allow orchestration of multiple clouds.
[ec2authtoken]	
allowed_auth_uris =	(ListOpt) Allowed keystone endpoints for auth_uri when multi_cloud is enabled. At least one endpoint needs to be specified.
auth_uri = None	(StrOpt) Authentication Endpoint URI.
multi_cloud = False	(BoolOpt) Allow orchestration of multiple clouds.
[heat_api]	
backlog = 4096	(IntOpt) Number of backlog requests to configure the socket with.
bind_host = 0.0.0.0	(StrOpt) Address to bind the server. Useful when selecting a particular network interface.
bind_port = 8004	(IntOpt) The port on which the server will listen.
cert_file = None	(StrOpt) Location of the SSL certificate file to use for SSL mode.
key_file = None	(StrOpt) Location of the SSL key file to use for enabling SSL mode.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
workers = 0	(IntOpt) Number of workers for Heat service.
[paste_deploy]	
api_paste_config = api-paste.ini	(StrOpt) The API paste config file to use.
flavor = None	(StrOpt) The flavor to use.
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients

Configuration option = Default value	Description
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 9.11. Description of configuration options for cfn_api

Configuration option = Default value	Description
[DEFAULT]	
instance_connection_https_validate_certificates = 1	(StrOpt) Instance connection to CFN/CW API validate certs if SSL is used.
instance_connection_is_secure = 0	(StrOpt) Instance connection to CFN/CW API via https.
[heat_api_cfn]	
backlog = 4096	(IntOpt) Number of backlog requests to configure the socket with.
bind_host = 0.0.0.0	(StrOpt) Address to bind the server. Useful when selecting a particular network interface.
bind_port = 8000	(IntOpt) The port on which the server will listen.
cert_file = None	(StrOpt) Location of the SSL certificate file to use for SSL mode.
key_file = None	(StrOpt) Location of the SSL key file to use for enabling SSL mode.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
workers = 0	(IntOpt) Number of workers for Heat service.
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 9.12. Description of configuration options for cloudwatch_api

Configuration option = Default value	Description
[DEFAULT]	
heat_watch_server_url =	(StrOpt) URL of the Heat CloudWatch server.
[heat_api_cloudwatch]	
backlog = 4096	(IntOpt) Number of backlog requests to configure the socket with.
bind_host = 0.0.0.0	(StrOpt) Address to bind the server. Useful when selecting a particular network interface.
bind_port = 8003	(IntOpt) The port on which the server will listen.
cert_file = None	(StrOpt) Location of the SSL certificate file to use for SSL mode.
key_file = None	(StrOpt) Location of the SSL key file to use for enabling SSL mode.
max_header_line = 16384	(IntOpt) Maximum line size of message headers to be accepted. max_header_line may need to be increased

Configuration option = Default value	Description
	when using large tokens (typically those generated by the Keystone v3 API with big service catalogs.)
workers = 0	(IntOpt) Number of workers for Heat service.
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 9.13. Description of configuration options for metadata_api

Configuration option = Default value	Description
[DEFAULT]	
heat_metadata_server_url =	(StrOpt) URL of the Heat metadata server.

Table 9.14. Description of configuration options for waitcondition_api

Configuration option = Default value	Description
[DEFAULT]	
heat_waitcondition_server_url =	(StrOpt) URL of the Heat waitcondition server.

Configure Clients

The following options allow configuration of the clients that Orchestration uses to talk to other services.

Table 9.15. Description of configuration options for clients

Configuration option = Default value	Description
[DEFAULT]	
region_name_for_services = None	(StrOpt) Default region name used to get services endpoints.
[clients]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.16. Description of configuration options for clients_backends

Configuration option = Default value	Description
[DEFAULT]	
cloud_backend = heat.engine.clients.OpenStackClients	(StrOpt) Fully qualified class name to use as a client backend.

Table 9.17. Description of configuration options for clients_ceilometer

Configuration option = Default value	Description
[clients_ceilometer]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.18. Description of configuration options for clients_cinder

Configuration option = Default value	Description
[clients_cinder]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.19. Description of configuration options for clients_heat

Configuration option = Default value	Description
[clients_heat]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
url = None	(StrOpt) Optional heat url in format like <code>http://0.0.0.0:8004/v1/%(tenant_id)s</code> .

Table 9.20. Description of configuration options for clients_keystone

Configuration option = Default value	Description
[clients_keystone]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.21. Description of configuration options for clients_neutron

Configuration option = Default value	Description
[clients_neutron]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.22. Description of configuration options for clients_nova

Configuration option = Default value	Description
[clients_nova]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.23. Description of configuration options for clients_swift

Configuration option = Default value	Description
[clients_swift]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Table 9.24. Description of configuration options for clients_trove

Configuration option = Default value	Description
[clients_trove]	
ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.

Configure the RPC messaging system

OpenStack projects use an open standard for messaging middleware known as AMQP. This messaging middleware enables the OpenStack services that run on multiple servers to talk to each other. OpenStack Oslo RPC supports three implementations of AMQP: RabbitMQ, Qpid, and ZeroMQ.

Configure RabbitMQ

OpenStack Oslo RPC uses RabbitMQ by default. Use these options to configure the RabbitMQ message system. The `rpc_backend` option is optional as long as RabbitMQ is the default messaging system. However, if it is included in the configuration, you must set it to `heat.openstack.common.rpc.impl_kombu`.

```
rpc_backend = heat.openstack.common.rpc.impl_kombu
```

Use these options to configure the RabbitMQ messaging system. You can configure messaging communication for different installation scenarios, tune retries for RabbitMQ, and define the size of the RPC thread pool. To monitor notifications through RabbitMQ, you must set the `notification_driver` option to `heat.notifier.rabbit_notifier` in the `heat.conf` file:

Table 9.25. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
<code>kombu_ssl_ca_certs =</code>	(StrOpt) SSL certification authority file (valid only if SSL enabled)
<code>kombu_ssl_certfile =</code>	(StrOpt) SSL cert file (valid only if SSL enabled)
<code>kombu_ssl_keyfile =</code>	(StrOpt) SSL key file (valid only if SSL enabled)
<code>kombu_ssl_version =</code>	(StrOpt) SSL version to use (valid only if SSL enabled). valid values are TLSv1, SSLv23 and SSLv3. SSLv2 may be available on some distributions
<code>rabbit_ha_queues = False</code>	(BoolOpt) use H/A queues in RabbitMQ (x-ha-policy: all). You need to wipe RabbitMQ database when changing this option.
<code>rabbit_host = localhost</code>	(StrOpt) The RabbitMQ broker address where a single node is used
<code>rabbit_hosts = \$rabbit_host:\$rabbit_port</code>	(ListOpt) RabbitMQ HA cluster host:port pairs
<code>rabbit_max_retries = 0</code>	(IntOpt) maximum retries with trying to connect to RabbitMQ (the default of 0 implies an infinite retry count)
<code>rabbit_password = guest</code>	(StrOpt) the RabbitMQ password
<code>rabbit_port = 5672</code>	(IntOpt) The RabbitMQ broker port where a single node is used
<code>rabbit_retry_backoff = 2</code>	(IntOpt) how long to backoff for between retries when connecting to RabbitMQ
<code>rabbit_retry_interval = 1</code>	(IntOpt) how frequently to retry connecting with RabbitMQ
<code>rabbit_use_ssl = False</code>	(BoolOpt) connect over SSL for RabbitMQ
<code>rabbit_userid = guest</code>	(StrOpt) the RabbitMQ userid
<code>rabbit_virtual_host = /</code>	(StrOpt) the RabbitMQ virtual host

Configure Qpid

Use these options to configure the Qpid messaging system for OpenStack Oslo RPC. Qpid is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `heat.conf` file:

```
rpc_backend=heat.openstack.common.rpc.impl_qpid
```

This critical option points the compute nodes to the Qpid broker (server). Set the `qpid_hostname` option to the host name where the broker runs in the `heat.conf` file.



Note

The `qpid_hostname` option accepts a host name or IP address value.

```
qpid_hostname = hostname.example.com
```

If the Qpid broker listens on a port other than the AMQP default of 5672, you must set the `qpid_port` option to that value:

```
qpid_port = 12345
```

If you configure the Qpid broker to require authentication, you must add a user name and password to the configuration:

```
qpid_username = username
qpid_password = password
```

By default, TCP is used as the transport. To enable SSL, set the `qpid_protocol` option:

```
qpid_protocol = ssl
```

Use these additional options to configure the Qpid messaging driver for OpenStack Oslo RPC. These options are used infrequently.

Table 9.26. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
<code>qpid_heartbeat = 60</code>	(IntOpt) Seconds between connection keepalive heartbeats
<code>qpid_hostname = localhost</code>	(StrOpt) Qpid broker hostname
<code>qpid_hosts = \$qpid_hostname:\$qpid_port</code>	(ListOpt) Qpid HA cluster host:port pairs
<code>qpid_password =</code>	(StrOpt) Password for qpid connection
<code>qpid_port = 5672</code>	(IntOpt) Qpid broker port
<code>qpid_protocol = tcp</code>	(StrOpt) Transport to use, either 'tcp' or 'ssl'
<code>qpid_sasl_mechanisms =</code>	(StrOpt) Space separated list of SASL mechanisms to use for auth
<code>qpid_tcp_nodelay = True</code>	(BoolOpt) Disable Nagle algorithm
<code>qpid_topology_version = 1</code>	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
<code>qpid_username =</code>	(StrOpt) Username for qpid connection

Configure ZeroMQ

Use these options to configure the ZeroMQ messaging system for OpenStack Oslo RPC. ZeroMQ is not the default messaging system, so you must enable it by setting the `rpc_backend` option in the `heat.conf` file:

Table 9.27. Description of configuration options for zeromq

Configuration option = Default value	Description
[DEFAULT]	
<code>rpc_zmq_bind_address = *</code>	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
<code>rpc_zmq_contexts = 1</code>	(IntOpt) Number of ZeroMQ contexts, defaults to 1
<code>rpc_zmq_host = oslo</code>	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
<code>rpc_zmq_ipc_dir = /var/run/openstack</code>	(StrOpt) Directory for holding IPC sockets
<code>rpc_zmq_matchmaker = heat.openstack.common.rpc.matchmaker.MatchMakerLocalhost</code>	(StrOpt) MatchMaker driver
<code>rpc_zmq_port = 9501</code>	(IntOpt) ZeroMQ receiver listening port
<code>rpc_zmq_topic_backlog = None</code>	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.

Configure messaging

Use these common options to configure the RabbitMQ, Qpid, and ZeroMq messaging drivers:

Table 9.28. Description of configuration options for amqp

Configuration option = Default value	Description
[DEFAULT]	
<code>amqp_auto_delete = False</code>	(BoolOpt) Auto-delete queues in amqp.
<code>amqp_durable_queues = False</code>	(BoolOpt) Use durable queues in amqp.
<code>control_exchange = heat</code>	(StrOpt) AMQP exchange to connect to if using RabbitMQ or Qpid
<code>default_notification_level = INFO</code>	(StrOpt) Default notification level for outgoing notifications
<code>default_publisher_id = None</code>	(StrOpt) Default publisher_id for outgoing notifications
<code>fake_rabbit = False</code>	(BoolOpt) If passed, use a fake RabbitMQ provider
<code>list_notifier_drivers = ['heat.openstack.common.notifier.no_op_notifier']</code>	(MultiStrOpt) List of drivers to send notifications
<code>notification_driver = []</code>	(MultiStrOpt) Driver or drivers to handle sending notifications
<code>notification_topics = notifications</code>	(ListOpt) AMQP topic used for OpenStack notifications

Table 9.29. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
<code>allowed_rpc_exception_modules = nova.exception, cinder.exception, exceptions</code>	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.

Configuration option = Default value	Description
engine_life_check_timeout = 2	(IntOpt) RPC timeout for the engine liveness check that is used for stack locking.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
rpc_backend = heat.openstack.common.rpc.impl_kombu	(StrOpt) The messaging module to use, defaults to kombu.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from call or multical
rpc_thread_pool_size = 64	(IntOpt) Size of RPC thread pool
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON)
[rpc_notifier2]	
topics = notifications	(ListOpt) AMQP topic(s) used for OpenStack notifications

Table 9.30. Description of configuration options for notification

Configuration option = Default value	Description
[DEFAULT]	
onready = None	(StrOpt) onready allows you to send a notification when the heat processes are ready to serve. This is either a module with the notify() method or a shell command. To enable notifications with systemd, one may use the 'systemd-notify --ready' shell command or the 'heat.common.systemd' notification module.
publish_errors = False	(BoolOpt) publish error events

New, updated and deprecated options for heat

Table 9.31. New options

Option = default value	(Type) Help string
engine_life_check_timeout = 2	(IntOpt) RPC timeout for the engine liveness check that is used for stack locking.
keystone_backend = heat.common.heat_keystoneclient.KeystoneClientV3	(StrOpt) Fully qualified class name to use as a keystone backend.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
onready = None	(StrOpt) onready allows you to send a notification when the heat processes are ready to serve. This is either a module with the notify() method or a shell command. To enable notifications with systemd, one may use the 'systemd-notify --ready' shell command or the 'heat.common.systemd' notification module.
region_name_for_services = None	(StrOpt) Default region name used to get services endpoints.
secure_proxy_ssl_header = X-Forwarded-Proto	(StrOpt) The HTTP Header that will be used to determine which the original request protocol scheme was, even if it was removed by an SSL terminator proxy.

Option = default value	(Type) Help string
stack_action_timeout = 3600	(IntOpt) Timeout in seconds for stack action (ie. create or update).
stack_domain_admin = None	(StrOpt) Keystone username, a user with roles sufficient to manage users and projects in the stack_user_domain.
stack_domain_admin_password = None	(StrOpt) Keystone password for stack_domain_admin user.
stack_user_domain = None	(StrOpt) Keystone domain ID which contains heat template-defined users.
clients/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_ceilometer/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_ceilometer/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_ceilometer/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_ceilometer/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_ceilometer/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_cinder/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_cinder/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_cinder/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_cinder/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_cinder/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_heat/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_heat/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_heat/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_heat/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_heat/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_heat/url = None	(StrOpt) Optional heat url in format like http://0.0.0.0:8004/v1/(tenant_id)s .
clients_keystone/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_keystone/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_keystone/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_keystone/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_keystone/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_neutron/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.

Option = default value	(Type) Help string
clients_neutron/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_neutron/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_neutron/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_neutron/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_nova/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_nova/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_nova/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_nova/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_nova/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_swift/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_swift/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_swift/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_swift/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_swift/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
clients_trove/ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
clients_trove/cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
clients_trove/endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
clients_trove/insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
clients_trove/key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
revision/heat_revision = unknown	(StrOpt) Heat build revision. If you would prefer to manage your build revision separately, you can move this section to a different file and add it as another config option.

Table 9.32. New default values

Option	Previous default value	New default value
allowed_rpc_exception_modules	heat.openstack.common.exception, heat.common.exception, nova.exception, cinder.exception, exceptions	nova.exception, cinder.exception, exceptions
cloud_backend	None	heat.engine.clients.OpenStackClients
control_exchange	openstack	heat
default_log_levels	amqpplib=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, eventlet.wsgi.server=WARN	amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%request_id)s %(user_identity)s] %(instance)s%(message)s

Table 9.33. Deprecated options

Deprecated option	New Option
db_backend	database/backend
log_config	log_config_append
sql_connection	database/connection
sql_idle_timeout	database/idle_timeout

10. Telemetry

Table of Contents

Telemetry sample configuration files	509
New, updated and deprecated options for ceilometer	529

The Telemetry service collects measurements within OpenStack. Its various agents and services are configured in the `/etc/ceilometer/ceilometer.conf` file.

To install Telemetry, see the *OpenStack Installation Guide* for your distribution (docs.openstack.org).

The following tables provide a comprehensive list of the Telemetry configuration options.

Table 10.1. Description of configuration options for alarm

Configuration option = Default value	Description
[alarm]	
evaluation_interval = 60	(IntOpt) Period of evaluation cycle, should be >= than configured pipeline interval for collection of underlying metrics.
evaluation_service = ceilometer.alarm.service.SingletonAlarmService	(StrOpt) Class to launch as alarm evaluation service.
notifier_rpc_topic = alarm_notifier	(StrOpt) The topic that ceilometer uses for alarm notifier messages.
partition_rpc_topic = alarm_partition_coordination	(StrOpt) The topic that ceilometer uses for alarm partition coordination messages.
record_history = True	(BoolOpt) Record alarm change events.
rest_notifier_certificate_file =	(StrOpt) SSL Client certificate for REST notifier.
rest_notifier_certificate_key =	(StrOpt) SSL Client private key for REST notifier.
rest_notifier_ssl_verify = True	(BoolOpt) Whether to verify the SSL Server certificate when calling alarm action.

Table 10.2. Description of configuration options for amqp

Configuration option = Default value	Description
[DEFAULT]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in amqp.
amqp_durable_queues = False	(BoolOpt) Use durable queues in amqp.
notification_driver = []	(MultiStrOpt) Driver or drivers to handle sending notifications
notification_topics = notifications	(ListOpt) AMQP topic used for OpenStack notifications

Table 10.3. Description of configuration options for api

Configuration option = Default value	Description
[DEFAULT]	
enable_v1_api = True	(BoolOpt) Deploy the deprecated v1 API.
max_request_body_size = 114688	(IntOpt) The maximum body size per request, in bytes

Configuration option = Default value	Description
pipeline_cfg_file = pipeline.yaml	(StrOpt) Configuration file for pipeline definition.
policy_default_rule = default	(StrOpt) Rule enforced when requested rule is not found
policy_file = policy.json	(StrOpt) JSON file containing policy
reserved_metadata_length = 256	(IntOpt) Limit on length of reserved metadata values.
reserved_metadata_namespace = metering.	(ListOpt) List of metadata prefixes reserved for metering use.
[api]	
host = 0.0.0.0	(StrOpt) The listen IP for the ceilometer API server.
port = 8777	(IntOpt) The port for the ceilometer API server.

Table 10.4. Description of configuration options for auth

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = keystone	(StrOpt) The strategy to use for auth: noauth or keystone.
[keystone_authtoken]	
admin_password = None	(StrOpt) Keystone account password
admin_tenant_name = admin	(StrOpt) Keystone service account tenant name to validate user tokens
admin_token = None	(StrOpt) Single shared secret with the Keystone configuration used for bootstrapping a Keystone installation, or otherwise bypassing the normal authentication process.
admin_user = None	(StrOpt) Keystone account username
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path
auth_host = 127.0.0.1	(StrOpt) Host providing the admin Identity API endpoint
auth_port = 35357	(IntOpt) Port of the admin Identity API endpoint
auth_protocol = https	(StrOpt) Protocol of the admin Identity API endpoint(http or https)
auth_uri = None	(StrOpt) Complete public Identity API endpoint
auth_version = None	(StrOpt) API version of the admin Identity API endpoint
cache = None	(StrOpt) Env key for the swift cache
cafile = None	(StrOpt) A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	(StrOpt) Required if Keystone server requires client certificate
delay_auth_decision = False	(BoolOpt) Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components
enforce_token_bind = permissive	(StrOpt) 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.
http_connect_timeout = None	(BoolOpt) Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.

Configuration option = Default value	Description
include_service_catalog = True	(BoolOpt) (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.
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) Required if Keystone server requires client certificate
memcache_secret_key = None	(StrOpt) (optional, mandatory if memcache_security_strategy is defined) this string is used for key derivation.
memcache_security_strategy = None	(StrOpt) (optional) if defined, indicate whether token data should be authenticated or authenticated and encrypted. Acceptable values are MAC or ENCRYPT. 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.
memcached_servers = None	(ListOpt) Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
revocation_cache_time = 300	(IntOpt) Determines the frequency at which the list of revoked tokens is retrieved from the Identity service (in seconds). A high number of revocation events combined with a low cache duration may significantly reduce performance.
signing_dir = None	(StrOpt) Directory used to cache files related to PKI tokens
token_cache_time = 300	(IntOpt) 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.
[service_credentials]	
insecure = False	(BoolOpt) Disables X.509 certificate validation when an SSL connection to Identity Service is established.
os_auth_url = http://localhost:5000/v2.0	(StrOpt) Auth URL to use for OpenStack service access.
os_cacert = None	(StrOpt) Certificate chain for SSL validation.
os_endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with OpenStack services.
os_password = admin	(StrOpt) Password to use for OpenStack service access.
os_region_name = None	(StrOpt) Region name to use for OpenStack service endpoints.
os_tenant_id =	(StrOpt) Tenant ID to use for OpenStack service access.
os_tenant_name = admin	(StrOpt) Tenant name to use for OpenStack service access.
os_username = ceilometer	(StrOpt) User name to use for OpenStack service access.

Table 10.5. Description of configuration options for cells

Configuration option = Default value	Description
[cells]	
bandwidth_update_interval = 600	(IntOpt) Seconds between bandwidth updates for cells.
call_timeout = 60	(IntOpt) Seconds to wait for response from a call to a cell.
capabilities = hypervisor=xenserver;kvm, os=linux;windows	(ListOpt) Key/Multi-value list with the capabilities of the cell
cell_type = compute	(StrOpt) Type of cell: api or compute

Configuration option = Default value	Description
enable = False	(BoolOpt) Enable cell functionality
manager = nova.cells.manager.CellsManager	(StrOpt) Manager for cells
mute_child_interval = 300	(IntOpt) Number of seconds after which a lack of capability and capacity updates signals the child cell is to be treated as a mute.
name = nova	(StrOpt) Name of this cell
reserve_percent = 10.0	(FloatOpt) Percentage of cell capacity to hold in reserve. Affects both memory and disk utilization
topic = cells	(StrOpt) The topic cells nodes listen on

Table 10.6. Description of configuration options for collector

Configuration option = Default value	Description
[DEFAULT]	
collector_workers = 1	(IntOpt) Number of workers for collector service. A single collector is enabled by default.
[collector]	
udp_address = 0.0.0.0	(StrOpt) Address to which the UDP socket is bound. Set to an empty string to disable.
udp_port = 4952	(IntOpt) Port to which the UDP socket is bound.
[dispatcher_file]	
backup_count = 0	(IntOpt) The max number of the files to keep.
file_path = None	(StrOpt) Name and the location of the file to record meters.
max_bytes = 0	(IntOpt) The max size of the file.

Table 10.7. Description of configuration options for common

Configuration option = Default value	Description
[DEFAULT]	
backdoor_port = None	(StrOpt) 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 service's log file.
disable_process_locking = False	(BoolOpt) Whether to disable inter-process locks.
fatal_deprecations = False	(BoolOpt) Make deprecations fatal
lock_path = None	(StrOpt) Directory to use for lock files.
memcached_servers = None	(ListOpt) Memcached servers or None for in process cache.
notification_workers = 1	(IntOpt) Number of workers for notification service. A single notification agent is enabled by default.

Table 10.8. Description of configuration options for database

Configuration option = Default value	Description
[DEFAULT]	
database_connection = None	(StrOpt) DEPRECATED - Database connection string.
mysql_engine = InnoDB	(StrOpt) MySQL engine to use.

Configuration option = Default value	Description
sqlite_db = ceilometer.sqlite	(StrOpt) The file name to use with SQLite
sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode
[database]	
backend = sqlalchemy	(StrOpt) The backend to use for db
connection = sqlite:///usr/lib/python/site-packages/ceilometer/openstack/common/db/\$sqlite_db	(StrOpt) The SQLAlchemy connection string used to connect to the database
connection_debug = 0	(IntOpt) Verbosity of SQL debugging information. 0=None, 100=Everything
connection_trace = False	(BoolOpt) Add python stack traces to SQL as comment strings
idle_timeout = 3600	(IntOpt) Timeout before idle sql connections are reaped
max_overflow = None	(IntOpt) If set, use this value for max_overflow with sqlalchemy
max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool
max_retries = 10	(IntOpt) Maximum db connection retries during startup. (setting -1 implies an infinite retry count)
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool
pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with sqlalchemy
retry_interval = 10	(IntOpt) Interval between retries of opening a sql connection
slave_connection =	(StrOpt) The SQLAlchemy connection string used to connect to the slave database
time_to_live = -1	(IntOpt) Number of seconds that samples are kept in the database for (<= 0 means forever).

Table 10.9. Description of configuration options for events

Configuration option = Default value	Description
[event]	
definitions_cfg_file = event_definitions.yaml	(StrOpt) Configuration file for event definitions.
drop_unmatched_notifications = False	(BoolOpt) Drop notifications if no event definition matches. (Otherwise, we convert them with just the default traits)
[notification]	
ack_on_event_error = True	(BoolOpt) Acknowledge message when event persistence fails.
store_events = False	(BoolOpt) Save event details.

Table 10.10. Description of configuration options for exchange

Configuration option = Default value	Description
[DEFAULT]	
cinder_control_exchange = cinder	(StrOpt) Exchange name for Cinder notifications.
control_exchange = openstack	(StrOpt) AMQP exchange to connect to if using RabbitMQ or Qpid
default_publisher_id = None	(StrOpt) Default publisher_id for outgoing notifications
glance_control_exchange = glance	(StrOpt) Exchange name for Glance notifications.
heat_control_exchange = heat	(StrOpt) Exchange name for Heat notifications

Configuration option = Default value	Description
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s %(message)s	(StrOpt) Format string to use for log messages without context
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d TRACE %(name)s %(instance)s	(StrOpt) Prefix each line of exception output with this format
publish_errors = False	(BoolOpt) Publish error events
syslog_log_facility = LOG_USER	(StrOpt) Syslog facility to receive log lines
use_stderr = True	(BoolOpt) Log output to standard error
use_syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and then will be changed in J to honor RFC5424
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).
[audit]	
api_audit_map = api_audit_map.conf	(StrOpt) File containing mapping for api paths and service endpoints
namespace = openstack	(StrOpt) namespace prefix for generated id

Table 10.13. Description of configuration options for qpid

Configuration option = Default value	Description
[DEFAULT]	
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats
qpid_hostname = localhost	(StrOpt) Qpid broker hostname
qpid_hosts = \$qpid_hostname:\$qpid_port	(ListOpt) Qpid HA cluster host:port pairs
qpid_password =	(StrOpt) Password for qpid connection
qpid_port = 5672	(IntOpt) Qpid broker port
qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'
qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth
qpid_tcp_nodelay = True	(BoolOpt) Disable Nagle algorithm
qpid_topology_version = 1	(IntOpt) The qpid topology version to use. Version 1 is what was originally used by impl_qpid. Version 2 includes some backwards-incompatible changes that allow broker federation to work. Users should update to version 2 when they are able to take everything down, as it requires a clean break.
qpid_username =	(StrOpt) Username for qpid connection

Table 10.14. Description of configuration options for rabbitmq

Configuration option = Default value	Description
[DEFAULT]	
fake_rabbit = False	(BoolOpt) If passed, use a fake RabbitMQ provider
kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled)

Configuration option = Default value	Description
kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled)
kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled)
kombu_ssl_version =	(StrOpt) If SSL is enabled, the SSL version to use. Valid values are TLSv1, SSLv23 and SSLv3. SSLv2 might be available on some distributions.
rabbit_ha_queues = False	(BoolOpt) Use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database.
rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used
rabbit_hosts = \$rabbit_host:\$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs
rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count)
rabbit_password = guest	(StrOpt) The RabbitMQ password
rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used
rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ
rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ
rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ
rabbit_userid = guest	(StrOpt) The RabbitMQ userid
rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host

Table 10.15. Description of configuration options for redis

Configuration option = Default value	Description
[DEFAULT]	
host = oslo	(StrOpt) Name of this node, which must be valid in an AMQP key. Can be an opaque identifier. For ZeroMQ only, must be a valid host name, FQDN, or IP address.
matchmaker_heartbeat_freq = 300	(IntOpt) Heartbeat frequency
matchmaker_heartbeat_ttl = 600	(IntOpt) Heartbeat time-to-live.
[matchmaker_redis]	
host = 127.0.0.1	(StrOpt) Host to locate redis
password = None	(StrOpt) Password for Redis server. (optional)
port = 6379	(IntOpt) Use this port to connect to redis host.
[matchmaker_ring]	
ringfile = /etc/oslo/matchmaker_ring.json	(StrOpt) Matchmaker ring file (JSON)

Table 10.16. Description of configuration options for rpc

Configuration option = Default value	Description
[DEFAULT]	
allowed_rpc_exception_modules = nova.exception, cinder.exception, exceptions	(ListOpt) Modules of exceptions that are permitted to be recreated upon receiving exception data from an rpc call.
dispatcher = ['database']	(MultiStrOpt) Dispatcher to process data.
rpc_backend = ceilometer.openstack.common.rpc.impl_kombu	(StrOpt) The messaging module to use, defaults to kombu.
rpc_cast_timeout = 30	(IntOpt) Seconds to wait before a cast expires (TTL). Only supported by impl_zmq.

Configuration option = Default value	Description
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool
rpc_response_timeout = 60	(IntOpt) Seconds to wait for a response from call or multical
rpc_thread_pool_size = 64	(IntOpt) Size of RPC thread pool
rpc_zmq_bind_address = *	(StrOpt) ZeroMQ bind address. Should be a wildcard (*), an ethernet interface, or IP. The "host" option should point or resolve to this address.
rpc_zmq_contexts = 1	(IntOpt) Number of ZeroMQ contexts, defaults to 1
rpc_zmq_host = oslo	(StrOpt) Name of this node. Must be a valid hostname, FQDN, or IP address. Must match "host" option, if running Nova.
rpc_zmq_ipc_dir = /var/run/openstack	(StrOpt) Directory for holding IPC sockets
rpc_zmq_matchmaker = ceilometer.openstack.common.rpc.matchmaker.MatchMakerLocalhost	(StrOpt) MatchMaker driver
rpc_zmq_port = 9501	(IntOpt) ZeroMQ receiver listening port
rpc_zmq_topic_backlog = None	(IntOpt) Maximum number of ingress messages to locally buffer per topic. Default is unlimited.
[publisher]	
metering_secret = change this or be hacked	(StrOpt) Secret value for signing metering messages.
[publisher_rpc]	
metering_topic = metering	(StrOpt) The topic that ceilometer uses for metering messages.
[rpc_notifier2]	
topics = notifications	(ListOpt) AMQP topic(s) used for OpenStack notifications

Table 10.17. Description of configuration options for ssl

Configuration option = Default value	Description
[ssl]	
ca_file = None	(StrOpt) CA certificate file to use to verify connecting clients
cert_file = None	(StrOpt) Certificate file to use when starting the server securely
key_file = None	(StrOpt) Private key file to use when starting the server securely

Table 10.18. Description of configuration options for swift

Configuration option = Default value	Description
[DEFAULT]	
reseller_prefix = AUTH_	(StrOpt) Swift reseller prefix. Must be on par with reseller_prefix in proxy-server.conf.

Table 10.19. Description of configuration options for vmware

Configuration option = Default value	Description
[vmware]	
api_retry_count = 10	(IntOpt) Number of times a VMware Vsphere API must be retried
host_ip =	(StrOpt) IP address of the VMware Vsphere host
host_password =	(StrOpt) Password of VMware Vsphere

Configuration option = Default value	Description
host_username =	(StrOpt) Username of VMware Vsphere
task_poll_interval = 0.5	(FloatOpt) Sleep time in seconds for polling an ongoing async task

Telemetry sample configuration files

All the files in this section can be found in the `/etc/ceilometer/` directory.

ceilometer.conf

The configuration for the Telemetry services and agents is found in the `ceilometer.conf` file.

This file must be modified after installation.

```
[DEFAULT]

#
# Options defined in ceilometer.middleware
#

# Exchanges name to listen for notifications. (multi valued)
#http_control_exchanges=nova
#http_control_exchanges=glance
#http_control_exchanges=neutron
#http_control_exchanges=cinder

#
# Options defined in ceilometer.pipeline
#

# Configuration file for pipeline definition. (string value)
#pipeline_cfg_file=pipeline.yaml

#
# Options defined in ceilometer.sample
#

# Source for samples emitted on this instance. (string value)
# Deprecated group/name - [DEFAULT]/counter_source
#sample_source=openstack

#
# Options defined in ceilometer.service
#

# Name of this node, which must be valid in an AMQP key. Can
# be an opaque identifier. For ZeroMQ only, must be a valid
# host name, FQDN, or IP address. (string value)
#host=ceilometer

# Dispatcher to process data. (multi valued)
```



```
#dispatcher=database

# Number of workers for collector service. A single
# collector is enabled by default. (integer value)
#collector_workers=1

# Number of workers for notification service. A single
# notification agent is enabled by default. (integer value)
#notification_workers=1

#
# Options defined in ceilometer.api.app
#

# The strategy to use for auth: noauth or keystone. (string
# value)
#auth_strategy=keystone

# Deploy the deprecated v1 API. (boolean value)
#enable_v1_api=true

#
# Options defined in ceilometer.compute.notifications
#

# Exchange name for Nova notifications. (string value)
#nova_control_exchange=nova

#
# Options defined in ceilometer.compute.util
#

# List of metadata prefixes reserved for metering use. (list
# value)
#reserved_metadata_namespace=metering.

# Limit on length of reserved metadata values. (integer value)
#reserved_metadata_length=256

#
# Options defined in ceilometer.compute.virt.inspector
#

# Inspector to use for inspecting the hypervisor layer.
# (string value)
#hypervisor_inspector=libvirt

#
# Options defined in ceilometer.compute.virt.libvirt.inspector
#

# Libvirt domain type (valid options are: kvm, lxc, qemu, uml,
# xen). (string value)
#libvirt_type=kvm
```



```
# Whether to disable inter-process locks. (boolean value)
#disable_process_locking=false

# Directory to use for lock files. (string value)
#lock_path=<None>

#
# Options defined in ceilometer.openstack.common.log
#

# Print debugging output (set logging level to DEBUG instead
# of default WARNING level). (boolean value)
#debug=false

# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
#verbose=false

# Log output to standard error (boolean value)
#use_stderr=true

# Format string to use for log messages with context (string
# value)
#logging_context_format_string=%(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 without context
# (string value)
#logging_default_format_string=%(asctime)s.%(msecs)03d %(process)d
%(levelname)s %(name)s [-] %(instance)s%(message)s

# Data to append to log format when level is DEBUG (string
# value)
#logging_debug_format_suffix=%(funcName)s %(pathname)s:%(lineno)d

# Prefix each line of exception output with this format
# (string value)
#logging_exception_prefix=%(asctime)s.%(msecs)03d %(process)d TRACE %(name)s
%(instance)s

# List of logger=LEVEL pairs (list value)
#default_log_levels=amqp=WARN,amqpplib=WARN,boto=WARN,qpidd=WARN,sqlalchemy=
WARN,suds=INFO,iso8601=WARN,requests.packages.urllib3.connectionpool=WARN

# Publish error events (boolean value)
#publish_errors=false

# Make deprecations fatal (boolean value)
#fatal_deprecations=false

# If an instance is passed with the log message, format it
# like this (string value)
#instance_format="[instance: %(uuid)s] "

# If an instance UUID is passed with the log message, format
# it like this (string value)
#instance_uuid_format="[instance: %(uuid)s] "
```

```
# The name of logging configuration file. It does not disable
# existing loggers, but just appends specified logging
# configuration to any other existing logging options. Please
# see the Python logging module documentation for details on
# logging configuration files. (string value)
# Deprecated group/name - [DEFAULT]/log_config
#log_config_append=<None>

# DEPRECATED. A logging.Formatter log message format string
# which may use any of the available logging.LogRecord
# attributes. This option is deprecated. Please use
# logging_context_format_string and
# logging_default_format_string instead. (string value)
#log_format=<None>

# Format string for %(asctime)s in log records. Default:
# %(default)s (string value)
#log_date_format=%Y-%m-%d %H:%M:%S

# (Optional) Name of log file to output to. If no default is
# set, logging will go to stdout. (string value)
# Deprecated group/name - [DEFAULT]/logfile
#log_file=<None>

# (Optional) The base directory used for relative --log-file
# paths (string value)
# Deprecated group/name - [DEFAULT]/logdir
#log_dir=<None>

# Use syslog for logging. Existing syslog format is DEPRECATED
# during I, and then will be changed in J to honor RFC5424
# (boolean value)
#use_syslog=false

# (Optional) Use syslog rfc5424 format for logging. If
# enabled, will add APP-NAME (RFC5424) before the MSG part of
# the syslog message. The old format without APP-NAME is
# deprecated in I, and will be removed in J. (boolean value)
#use_syslog_rfc_format=false

# Syslog facility to receive log lines (string value)
#syslog_log_facility=LOG_USER

#
# Options defined in ceilometer.openstack.common.middleware.sizelimit
#

# The maximum body size per request, in bytes (integer value)
# Deprecated group/name - [DEFAULT]/osapi_max_request_body_size
#max_request_body_size=114688

#
# Options defined in ceilometer.openstack.common.notifier.api
#

# Driver or drivers to handle sending notifications (multi
# valued)
```

```
#notification_driver=

# Default notification level for outgoing notifications
# (string value)
#default_notification_level=INFO

# Default publisher_id for outgoing notifications (string
# value)
#default_publisher_id=<None>

#
# Options defined in ceilometer.openstack.common.notifier.rpc_notifier
#

# AMQP topic used for OpenStack notifications (list value)
#notification_topics=notifications

#
# Options defined in ceilometer.openstack.common.policy
#

# JSON file containing policy (string value)
#policy_file=policy.json

# Rule enforced when requested rule is not found (string
# value)
#policy_default_rule=default

#
# Options defined in ceilometer.openstack.common.rpc
#

# The messaging module to use, defaults to kombu. (string
# value)
#rpc_backend=ceilometer.openstack.common.rpc.impl_kombu

# Size of RPC thread pool (integer value)
#rpc_thread_pool_size=64

# Size of RPC connection pool (integer value)
#rpc_conn_pool_size=30

# Seconds to wait for a response from call or multicall
# (integer value)
#rpc_response_timeout=60

# Seconds to wait before a cast expires (TTL). Only supported
# by impl_zmq. (integer value)
#rpc_cast_timeout=30

# Modules of exceptions that are permitted to be recreated
# upon receiving exception data from an rpc call. (list value)
#allowed_rpc_exception_modules=nova.exception,cinder.exception,exceptions

# If passed, use a fake RabbitMQ provider (boolean value)
#fake_rabbit=false
```

```
# AMQP exchange to connect to if using RabbitMQ or Qpid
# (string value)
#control_exchange=openstack

#
# Options defined in ceilometer.openstack.common.rpc.amqp
#

# Use durable queues in amqp. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_durable_queues
#amqp_durable_queues=false

# Auto-delete queues in amqp. (boolean value)
#amqp_auto_delete=false

#
# Options defined in ceilometer.openstack.common.rpc.impl_kombu
#

# If SSL is enabled, the SSL version to use. Valid values are
# TLSv1, SSLv23 and SSLv3. SSLv2 might be available on some
# distributions. (string value)
#kombu_ssl_version=

# SSL key file (valid only if SSL enabled) (string value)
#kombu_ssl_keyfile=

# SSL cert file (valid only if SSL enabled) (string value)
#kombu_ssl_certfile=

# SSL certification authority file (valid only if SSL enabled)
# (string value)
#kombu_ssl_ca_certs=

# The RabbitMQ broker address where a single node is used
# (string value)
#rabbit_host=localhost

# The RabbitMQ broker port where a single node is used
# (integer value)
#rabbit_port=5672

# RabbitMQ HA cluster host:port pairs (list value)
#rabbit_hosts=$rabbit_host:$rabbit_port

# Connect over SSL for RabbitMQ (boolean value)
#rabbit_use_ssl=false

# The RabbitMQ userid (string value)
#rabbit_userid=guest

# The RabbitMQ password (string value)
#rabbit_password=guest

# The RabbitMQ virtual host (string value)
#rabbit_virtual_host=/

# How frequently to retry connecting with RabbitMQ (integer
```

```
# value)
#rabbit_retry_interval=1

# How long to backoff for between retries when connecting to
# RabbitMQ (integer value)
#rabbit_retry_backoff=2

# Maximum number of RabbitMQ connection retries. Default is 0
# (infinite retry count) (integer value)
#rabbit_max_retries=0

# Use HA queues in RabbitMQ (x-ha-policy: all). If you change
# this option, you must wipe the RabbitMQ database. (boolean
# value)
#rabbit_ha_queues=false

#
# Options defined in ceilometer.openstack.common.rpc.impl_qpid
#

# Qpid broker hostname (string value)
#qpid_hostname=localhost

# Qpid broker port (integer value)
#qpid_port=5672

# Qpid HA cluster host:port pairs (list value)
#qpid_hosts=$qpid_hostname:$qpid_port

# Username for qpid connection (string value)
#qpid_username=

# Password for qpid connection (string value)
#qpid_password=

# Space separated list of SASL mechanisms to use for auth
# (string value)
#qpid_sasl_mechanisms=

# Seconds between connection keepalive heartbeats (integer
# value)
#qpid_heartbeat=60

# Transport to use, either 'tcp' or 'ssl' (string value)
#qpid_protocol=tcp

# Disable Nagle algorithm (boolean value)
#qpid_tcp_nodelay=true

# The qpid topology version to use. Version 1 is what was
# originally used by impl_qpid. Version 2 includes some
# backwards-incompatible changes that allow broker federation
# to work. Users should update to version 2 when they are
# able to take everything down, as it requires a clean break.
# (integer value)
#qpid_topology_version=1

#
```

```
# Options defined in ceilometer.openstack.common.rpc.impl_zmq
#

# ZeroMQ bind address. Should be a wildcard (*), an ethernet
# interface, or IP. The "host" option should point or resolve
# to this address. (string value)
#rpc_zmq_bind_address=*

# MatchMaker driver (string value)
#rpc_zmq_matchmaker=ceilometer.openstack.common.rpc.matchmaker.
MatchMakerLocalhost

# ZeroMQ receiver listening port (integer value)
#rpc_zmq_port=9501

# Number of ZeroMQ contexts, defaults to 1 (integer value)
#rpc_zmq_contexts=1

# Maximum number of ingress messages to locally buffer per
# topic. Default is unlimited. (integer value)
#rpc_zmq_topic_backlog=<None>

# Directory for holding IPC sockets (string value)
#rpc_zmq_ipc_dir=/var/run/openstack

# Name of this node. Must be a valid hostname, FQDN, or IP
# address. Must match "host" option, if running Nova. (string
# value)
#rpc_zmq_host=ceilometer

#
# Options defined in ceilometer.openstack.common.rpc.matchmaker
#

# Heartbeat frequency (integer value)
#matchmaker_heartbeat_freq=300

# Heartbeat time-to-live. (integer value)
#matchmaker_heartbeat_ttl=600

#
# Options defined in ceilometer.orchestration.notifications
#

# Exchange name for Heat notifications (string value)
#heat_control_exchange=heat

#
# Options defined in ceilometer.storage
#

# DEPRECATED - Database connection string. (string value)
#database_connection=<None>

#
# Options defined in ceilometer.storage.sqlalchemy.models
```



```
#  
  
# MySQL engine to use. (string value)  
#mysql_engine=InnoDB  
  
#  
# Options defined in ceilometer.volume.notifications  
#  
  
# Exchange name for Cinder notifications. (string value)  
#cinder_control_exchange=cinder  
  
[alarm]  
  
#  
# Options defined in ceilometer.cli  
#  
  
# Class to launch as alarm evaluation service. (string value)  
#evaluation_service=ceilometer.alarm.service.SingletonAlarmService  
  
#  
# Options defined in ceilometer.alarm.notifier.rest  
#  
  
# SSL Client certificate for REST notifier. (string value)  
#rest_notifier_certificate_file=  
  
# SSL Client private key for REST notifier. (string value)  
#rest_notifier_certificate_key=  
  
# Whether to verify the SSL Server certificate when calling  
# alarm action. (boolean value)  
#rest_notifier_ssl_verify=true  
  
#  
# Options defined in ceilometer.alarm.rpc  
#  
  
# The topic that ceilometer uses for alarm notifier messages.  
# (string value)  
#notifier_rpc_topic=alarm_notifier  
  
# The topic that ceilometer uses for alarm partition  
# coordination messages. (string value)  
#partition_rpc_topic=alarm_partition_coordination  
  
#  
# Options defined in ceilometer.alarm.service  
#  
  
# Period of evaluation cycle, should be >= than configured  
# pipeline interval for collection of underlying metrics.  
# (integer value)  
# Deprecated group/name - [alarm]/threshold_evaluation_interval
```

```
#evaluation_interval=60

#
# Options defined in ceilometer.api.controllers.v2
#

# Record alarm change events. (boolean value)
#record_history=true

[api]

#
# Options defined in ceilometer.api
#

# The port for the ceilometer API server. (integer value)
# Deprecated group/name - [DEFAULT]/metering_api_port
#port=8777

# The listen IP for the ceilometer API server. (string value)
#host=0.0.0.0

[collector]

#
# Options defined in ceilometer.collector
#

# Address to which the UDP socket is bound. Set to an empty
# string to disable. (string value)
#udp_address=0.0.0.0

# Port to which the UDP socket is bound. (integer value)
#udp_port=4952

[database]

#
# Options defined in ceilometer.openstack.common.db.api
#

# The backend to use for db (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend=sqlalchemy

#
# Options defined in ceilometer.openstack.common.db.sqlalchemy.session
#

# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
#connection=sqlite:///ceilometer/openstack/common/db/$sqlite_db
```

```
# The SQLAlchemy connection string used to connect to the
# slave database (string value)
#slave_connection=

# Timeout before idle sql connections are reaped (integer
# value)
# Deprecated group/name - [DEFAULT]/sql_idle_timeout
# Deprecated group/name - [DATABASE]/sql_idle_timeout
# Deprecated group/name - [sql]/idle_timeout
#idle_timeout=3600

# Minimum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_min_pool_size
# Deprecated group/name - [DATABASE]/sql_min_pool_size
#min_pool_size=1

# Maximum number of SQL connections to keep open in a pool
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_pool_size
# Deprecated group/name - [DATABASE]/sql_max_pool_size
#max_pool_size=<None>

# Maximum db connection retries during startup. (setting -1
# implies an infinite retry count) (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_retries
# Deprecated group/name - [DATABASE]/sql_max_retries
#max_retries=10

# Interval between retries of opening a sql connection
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_retry_interval
# Deprecated group/name - [DATABASE]/reconnect_interval
#retry_interval=10

# If set, use this value for max_overflow with sqlalchemy
# (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_overflow
# Deprecated group/name - [DATABASE]/sqlalchemy_max_overflow
#max_overflow=<None>

# Verbosity of SQL debugging information. 0=None,
# 100=Everything (integer value)
# Deprecated group/name - [DEFAULT]/sql_connection_debug
#connection_debug=0

# Add python stack traces to SQL as comment strings (boolean
# value)
# Deprecated group/name - [DEFAULT]/sql_connection_trace
#connection_trace=false

# If set, use this value for pool_timeout with sqlalchemy
# (integer value)
# Deprecated group/name - [DATABASE]/sqlalchemy_pool_timeout
#pool_timeout=<None>

#
# Options defined in ceilometer.storage
```

```
#
# Number of seconds that samples are kept in the database for
# (<= 0 means forever). (integer value)
#time_to_live=-1

[dispatcher_file]

#
# Options defined in ceilometer.dispatcher.file
#

# Name and the location of the file to record meters. (string
# value)
#file_path=<None>

# The max size of the file. (integer value)
#max_bytes=0

# The max number of the files to keep. (integer value)
#backup_count=0

[event]

#
# Options defined in ceilometer.event.converter
#

# Configuration file for event definitions. (string value)
#definitions_cfg_file=event_definitions.yaml

# Drop notifications if no event definition matches.
# (Otherwise, we convert them with just the default traits)
# (boolean value)
#drop_unmatched_notifications=false

[keystone_authtoken]

#
# Options defined in keystoneclient.middleware.auth_token
#

# Prefix to prepend at the beginning of the path (string
# value)
#auth_admin_prefix=

# Host providing the admin Identity API endpoint (string
# value)
#auth_host=127.0.0.1

# Port of the admin Identity API endpoint (integer value)
#auth_port=35357

# Protocol of the admin Identity API endpoint(http or https)
# (string value)
#auth_protocol=https
```

```
# Complete public Identity API endpoint (string value)
#auth_uri=<None>

# API version of the admin Identity API endpoint (string
# value)
#auth_version=<None>

# Do not handle authorization requests within the middleware,
# but delegate the authorization decision to downstream WSGI
# components (boolean value)
#delay_auth_decision=false

# Request timeout value for communicating with Identity API
# server. (boolean value)
#http_connect_timeout=<None>

# How many times are we trying to reconnect when communicating
# with Identity API Server. (integer value)
#http_request_max_retries=3

# Allows to pass in the name of a fake http_handler callback
# function used instead of httplib.HTTPConnection or
# httplib.HTTPSConnection. Useful for unit testing where
# network is not available. (string value)
#http_handler=<None>

# Single shared secret with the Keystone configuration used
# for bootstrapping a Keystone installation, or otherwise
# bypassing the normal authentication process. (string value)
#admin_token=<None>

# Keystone account username (string value)
#admin_user=<None>

# Keystone account password (string value)
#admin_password=<None>

# Keystone service account tenant name to validate user tokens
# (string value)
#admin_tenant_name=admin

# Env key for the swift cache (string value)
#cache=<None>

# Required if Keystone server requires client certificate
# (string value)
#certfile=<None>

# Required if Keystone server requires client certificate
# (string value)
#keyfile=<None>

# A PEM encoded Certificate Authority to use when verifying
# HTTPS connections. Defaults to system CAs. (string value)
#cafile=<None>

# Verify HTTPS connections. (boolean value)
#insecure=false

# Directory used to cache files related to PKI tokens (string
```

```
# value)
#signing_dir=<None>

# If defined, the memcache server(s) to use for caching (list
# value)
# Deprecated group/name - [DEFAULT]/memcache_servers
#memcached_servers=<None>

# In order to prevent excessive requests and validations, the
# middleware uses an in-memory cache for the tokens the
# Keystone API returns. This is only valid if memcache_servers
# is defined. Set to -1 to disable caching completely.
# (integer value)
#token_cache_time=300

# Value only used for unit testing (integer value)
#revocation_cache_time=1

# (optional) if defined, indicate whether token data should be
# authenticated or authenticated and encrypted. Acceptable
# values are MAC or ENCRYPT. 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. (string value)
#memcache_security_strategy=<None>

# (optional, mandatory if memcache_security_strategy is
# defined) this string is used for key derivation. (string
# value)
#memcache_secret_key=<None>

# (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. (boolean value)
#include_service_catalog=true

# 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. (string
# value)
#enforce_token_bind=permissive

[matchmaker_redis]

#
# Options defined in ceilometer.openstack.common.rpc.matchmaker_redis
#

# Host to locate redis (string value)
#host=127.0.0.1

# Use this port to connect to redis host. (integer value)
```

```
#port=6379

# Password for Redis server. (optional) (string value)
#password=<None>

[matchmaker_ring]

#
# Options defined in ceilometer.openstack.common.rpc.matchmaker_ring
#

# Matchmaker ring file (JSON) (string value)
# Deprecated group/name - [DEFAULT]/matchmaker_ringfile
#ringfile=/etc/oslo/matchmaker_ring.json

[notification]

#
# Options defined in ceilometer.notification
#

# Acknowledge message when event persistence fails. (boolean
# value)
#ack_on_event_error=true

# Save event details. (boolean value)
#store_events=false

[publisher]

#
# Options defined in ceilometer.publisher.utils
#

# Secret value for signing metering messages. (string value)
# Deprecated group/name - [DEFAULT]/metering_secret
# Deprecated group/name - [publisher_rpc]/metering_secret
#metering_secret=change this or be hacked

[publisher_rpc]

#
# Options defined in ceilometer.publisher.rpc
#

# The topic that ceilometer uses for metering messages.
# (string value)
#metering_topic=metering

[rpc_notifier2]

#
# Options defined in ceilometer.openstack.common.notifier.rpc_notifier2
#
```



```
#key_file=<None>

[vmware]

#
# Options defined in ceilometer.compute.virt.vmware.inspector
#

# IP address of the VMware Vsphere host (string value)
#host_ip=

# Username of VMware Vsphere (string value)
#host_username=

# Password of VMware Vsphere (string value)
#host_password=

# Number of times a VMware Vsphere API must be retried
# (integer value)
#api_retry_count=10

# Sleep time in seconds for polling an ongoing async task
# (floating point value)
#task_poll_interval=0.5
```

event_definitions.yaml

The `event_definitions.yaml` file defines how events received from other OpenStack components should be translated to Telemetry samples.

You should not need to modify this file.

```
---
- event_type: compute.instance.*
  traits: &instance_traits
    tenant_id:
      fields: payload.tenant_id
    user_id:
      fields: payload.user_id
    instance_id:
      fields: payload.instance_id
    host:
      fields: publisher_id
      plugin:
        name: split
        parameters:
          segment: 1
          max_split: 1
    service:
      fields: publisher_id
      plugin: split
    memory_mb:
      type: int
      fields: payload.memory_mb
    disk_gb:
```

```
    type: int
    fields: payload.disk_gb
  root_gb:
    type: int
    fields: payload.root_gb
  ephemeral_gb:
    type: int
    fields: payload.ephemeral_gb
  vcpus:
    type: int
    fields: payload.vcpus
  instance_type_id:
    type: int
    fields: payload.instance_type_id
  instance_type:
    fields: payload.instance_type
  state:
    fields: payload.state
  os_architecture:
    fields: payload.image_meta.'org.openstack_1__architecture'
  os_version:
    fields: payload.image_meta.'org.openstack_1__os_version'
  os_distro:
    fields: payload.image_meta.'org.openstack_1__os_distro'
  launched_at:
    type: datetime
    fields: payload.launched_at
  deleted_at:
    type: datetime
    fields: payload.deleted_at
- event_type: compute.instance.exists
traits:
  <<: *instance_traits
  audit_period_beginning:
    type: datetime
    fields: payload.audit_period_beginning
  audit_period_ending:
    type: datetime
    fields: payload.audit_period_ending
```

pipeline.yaml

Pipelines describe a coupling between sources of samples and the corresponding sinks for transformation and publication of these data. They are defined in the `pipeline.yaml` file.

You should not need to modify this file.

```
---
sources:
  - name: meter_source
    interval: 600
    meters:
      - "*"
    sinks:
      - meter_sink
  - name: cpu_source
    interval: 600
```



```

        name: "network\\. (incoming|outgoing)\\. (bytes|packets) "
        unit: "(B|packet)"
    target:
        map_to:
            name: "network.\\1.\\2.rate"
            unit: "\\1/s"
            type: "gauge"
    publishers:
        - rpc://

```

policy.json

The `policy.json` file defines additional access controls that apply to the Telemetry service.

```

{
  "context_is_admin": [{"role:admin"}]
}

```

New, updated and deprecated options for ceilometer

Table 10.20. New options

Option = default value	(Type) Help string
collector_workers = 1	(IntOpt) Number of workers for collector service. A single collector is enabled by default.
dispatcher = ['database']	(MultiStrOpt) Dispatcher to process data.
heat_control_exchange = heat	(StrOpt) Exchange name for Heat notifications
kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
log_config_append = None	(StrOpt) The name of logging configuration file. It does not disable existing loggers, but just appends specified logging configuration to any other existing logging options. Please see the Python logging module documentation for details on logging configuration files.
max_request_body_size = 114688	(IntOpt) The maximum body size per request, in bytes
notification_workers = 1	(IntOpt) Number of workers for notification service. A single notification agent is enabled by default.
use_syslog_rfc_format = False	(BoolOpt) (Optional) Use syslog rfc5424 format for logging. If enabled, will add APP-NAME (RFC5424) before the MSG part of the syslog message. The old format without APP-NAME is deprecated in I, and will be removed in J.
event/definitions_cfg_file = event_definitions.yaml	(StrOpt) Configuration file for event definitions.
event/drop_unmatched_notifications = False	(BoolOpt) Drop notifications if no event definition matches. (Otherwise, we convert them with just the default traits)
notification/ack_on_event_error = True	(BoolOpt) Acknowledge message when event persistence fails.
notification/store_events = False	(BoolOpt) Save event details.
publisher/metering_secret = change this or be hacked	(StrOpt) Secret value for signing metering messages.
vmware/api_retry_count = 10	(IntOpt) Number of times a VMware Vsphere API must be retried

Option = default value	(Type) Help string
vmware/host_ip =	(StrOpt) IP address of the VMware Vsphere host
vmware/host_password =	(StrOpt) Password of VMware Vsphere
vmware/host_username =	(StrOpt) Username of VMware Vsphere
vmware/task_poll_interval = 0.5	(FloatOpt) Sleep time in seconds for polling an ongoing async task

Table 10.21. New default values

Option	Previous default value	New default value
allowed_rpc_exception_modules	ceilometer.openstack.common.exceptions, nova.exception, cinder.exception, exceptions	nova.exception, cinder.exception, exceptions
default_log_levels	amqpplib=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, eventlet.wsgi.server=WARN	amqp=WARN, amqpplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN
logging_context_format_string	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user)s %(tenant)s] %(instance)s%(message)s	%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s
cells/cell_type	None	compute

Table 10.22. Deprecated options

Deprecated option	New Option
log_config	log_config_append
publisher_rpc/metering_secret	publisher/metering_secret

Appendix A. Firewalls and default ports

On some deployments, such as ones where restrictive firewalls are in place, you might need to manually configure a firewall to permit OpenStack service traffic.

To manually configure a firewall, you must permit traffic through the ports that each OpenStack service uses. This table lists the default ports that each OpenStack service uses:

Table A.1. Default ports that OpenStack components use

OpenStack service	Default ports	Port type
Block Storage (<code>cinder</code>)	8776	publicurl and adminurl
Compute (<code>nova</code>) endpoints	8774	publicurl and adminurl
Compute API (<code>nova-api</code>)	8773, 8775	
Compute ports for access to virtual machine consoles	5900-5999	
Compute VNC proxy for browsers (<code>openstack-nova-novncproxy</code>)	6080	
Compute VNC proxy for traditional VNC clients (<code>openstack-nova-xvncproxy</code>)	6081	
Proxy port for HTML5 console used by Compute service	6082	
Identity service (<code>keystone</code>) administrative endpoint	35357	adminurl
Identity service public endpoint	5000	publicurl
Image Service (<code>glance</code>) API	9292	publicurl and adminurl
Image Service registry	9191	
Networking (<code>neutron</code>)	9696	publicurl and adminurl
Object Storage (<code>swift</code>)	6000, 6001, 6002	
Orchestration (<code>heat</code>) endpoint	8004	publicurl and adminurl
Orchestration AWS CloudFormation-compatible API (<code>openstack-heat-api-cfn</code>)	8000	
Orchestration AWS CloudWatch-compatible API (<code>openstack-heat-api-cloudwatch</code>)	8003	
Telemetry (<code>ceilometer</code>)	8777	publicurl and adminurl

To function properly, some OpenStack components depend on other, non-OpenStack services. For example, the OpenStack dashboard uses HTTP for non-secure communication. In this case, you must configure the firewall to allow traffic to and from HTTP.

This table lists the ports that other OpenStack components use:

Table A.2. Default ports that secondary services related to OpenStack components use

Service	Default port	Used by
HTTP	80	OpenStack dashboard (<code>Horizon</code>) when it is not configured to use secure access.
HTTP alternate	8080	OpenStack Object Storage (<code>swift</code>) service.
HTTPS	443	Any OpenStack service that is enabled for SSL, especially secure-access dashboard.
rsync	873	OpenStack Object Storage. Required.
iSCSI target	3260	OpenStack Block Storage. Required.

Appendix B. Community support

Table of Contents

Documentation	533
ask.openstack.org	534
OpenStack mailing lists	534
The OpenStack wiki	535
The Launchpad Bugs area	535
The OpenStack IRC channel	536
Documentation feedback	536
OpenStack distribution packages	536

The following resources are available to help you run and use OpenStack. The OpenStack community constantly improves and adds to the main features of OpenStack, but if you have any questions, do not hesitate to ask. Use the following resources to get OpenStack support, and troubleshoot your installations.

Documentation

For the available OpenStack documentation, see docs.openstack.org.

To provide feedback on documentation, join and use the <openstack-docs@lists.openstack.org> mailing list at [OpenStack Documentation Mailing List](#), or [report a bug](#).

The following books explain how to install an OpenStack cloud and its associated components:

- [Installation Guide for Debian 7.0](#)
- [Installation Guide for openSUSE and SUSE Linux Enterprise Server](#)
- [Installation Guide for Red Hat Enterprise Linux, CentOS, and Fedora](#)
- [Installation Guide for Ubuntu 12.04/14.04 \(LTS\)](#)

The following books explain how to configure and run an OpenStack cloud:

- [Cloud Administrator Guide](#)
- [Configuration Reference](#)
- [Operations Guide](#)
- [High Availability Guide](#)
- [Security Guide](#)

- [Virtual Machine Image Guide](#)

The following books explain how to use the OpenStack dashboard and command-line clients:

- [API Quick Start](#)
- [End User Guide](#)
- [Admin User Guide](#)
- [Command-Line Interface Reference](#)

The following documentation provides reference and guidance information for the OpenStack APIs:

- [OpenStack API Complete Reference \(HTML\)](#)
- [API Complete Reference \(PDF\)](#)
- [OpenStack Block Storage Service API v2 Reference](#)
- [OpenStack Compute API v2 and Extensions Reference](#)
- [OpenStack Identity Service API v2.0 Reference](#)
- [OpenStack Image Service API v2 Reference](#)
- [OpenStack Networking API v2.0 Reference](#)
- [OpenStack Object Storage API v1 Reference](#)

The [Training Guides](#) offer software training for cloud administration and management.

ask.openstack.org

During the set up or testing of OpenStack, you might have questions about how a specific task is completed or be in a situation where a feature does not work correctly. Use the ask.openstack.org site to ask questions and get answers. When you visit the <http://ask.openstack.org> site, scan the recently asked questions to see whether your question has already been answered. If not, ask a new question. Be sure to give a clear, concise summary in the title and provide as much detail as possible in the description. Paste in your command output or stack traces, links to screen shots, and any other information which might be useful.

OpenStack mailing lists

A great way to get answers and insights is to post your question or problematic scenario to the OpenStack mailing list. You can learn from and help others who might have similar issues. To subscribe or view the archives, go to <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack>. You might be interested in the other mailing lists for specific projects or development, which you can find [on the wiki](#). A description of all mailing lists is available at <http://wiki.openstack.org/MailingLists>.

The OpenStack wiki

The [OpenStack wiki](#) contains a broad range of topics but some of the information can be difficult to find or is a few pages deep. Fortunately, the wiki search feature enables you to search by title or content. If you search for specific information, such as about networking or nova, you can find a large amount of relevant material. More is being added all the time, so be sure to check back often. You can find the search box in the upper-right corner of any OpenStack wiki page.

The Launchpad Bugs area

The OpenStack community values your set up and testing efforts and wants your feedback. To log a bug, you must sign up for a Launchpad account at <https://launchpad.net/+login>. You can view existing bugs and report bugs in the Launchpad Bugs area. Use the search feature to determine whether the bug has already been reported or already been fixed. If it still seems like your bug is unreported, fill out a bug report.

Some tips:

- Give a clear, concise summary.
- Provide as much detail as possible in the description. Paste in your command output or stack traces, links to screen shots, and any other information which might be useful.
- Be sure to include the software and package versions that you are using, especially if you are using a development branch, such as, "Juno release" vs `git commit bc79c3ecc55929bac585d04a03475b72e06a3208`.
- Any deployment-specific information is helpful, such as whether you are using Ubuntu 14.04 or are performing a multi-node installation.

The following Launchpad Bugs areas are available:

- [Bugs: OpenStack Block Storage \(cinder\)](#)
- [Bugs: OpenStack Compute \(nova\)](#)
- [Bugs: OpenStack Dashboard \(horizon\)](#)
- [Bugs: OpenStack Identity \(keystone\)](#)
- [Bugs: OpenStack Image Service \(glance\)](#)
- [Bugs: OpenStack Networking \(neutron\)](#)
- [Bugs: OpenStack Object Storage \(swift\)](#)
- [Bugs: Bare Metal \(ironic\)](#)
- [Bugs: Data Processing Service \(sahara\)](#)
- [Bugs: Database Service \(trove\)](#)

- [Bugs: Orchestration \(heat\)](#)
- [Bugs: Telemetry \(ceilometer\)](#)
- [Bugs: Queue Service \(marconi\)](#)
- [Bugs: OpenStack API Documentation \(api.openstack.org\)](#)
- [Bugs: OpenStack Documentation \(docs.openstack.org\)](#)

The OpenStack IRC channel

The OpenStack community lives in the #openstack IRC channel on the Freenode network. You can hang out, ask questions, or get immediate feedback for urgent and pressing issues. To install an IRC client or use a browser-based client, go to <http://webchat.freenode.net/>. You can also use Colloquy (Mac OS X, <http://colloquy.info/>), mIRC (Windows, <http://www.mirc.com/>), or XChat (Linux). When you are in the IRC channel and want to share code or command output, the generally accepted method is to use a Paste Bin. The OpenStack project has one at <http://paste.openstack.org>. Just paste your longer amounts of text or logs in the web form and you get a URL that you can paste into the channel. The OpenStack IRC channel is #openstack on irc.freenode.net. You can find a list of all OpenStack IRC channels at <https://wiki.openstack.org/wiki/IRC>.

Documentation feedback

To provide feedback on documentation, join and use the <openstack-docs@lists.openstack.org> mailing list at [OpenStack Documentation Mailing List](#), or [report a bug](#).

OpenStack distribution packages

The following Linux distributions provide community-supported packages for OpenStack:

- **Debian:** <http://wiki.debian.org/OpenStack>
- **CentOS, Fedora, and Red Hat Enterprise Linux:** <http://openstack.redhat.com/>
- **openSUSE and SUSE Linux Enterprise Server:** <http://en.opensuse.org/Portal:OpenStack>
- **Ubuntu:** <https://wiki.ubuntu.com/ServerTeam/CloudArchive>