OpenStack

Configuration Reference

liberty (June 30, 2015)

DRAFT

Liberty

docs.openstack.org

OpenStack Configuration Reference

liberty (2015-06-30)

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.

Table of Contents

List of Figures

List of Tables

List of Examples

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:

- Bare metal service
- OpenStack Block Storage
- OpenStack Compute
- OpenStack dashboard
- Database service for OpenStack
- Data processing service
- OpenStack Identity
- OpenStack Image service
- OpenStack Networking
- OpenStack Object Storage
- Orchestration
- Telemetry

Conventions

The OpenStack documentation uses several typesetting conventions.

Notices

Notices take these forms:

Note

A handy tip or reminder.

Important

Something you must be aware of before proceeding.

Warning

Critical information about the risk of data loss or security issues.

Command prompts

$ prompt Any user, including the `root` user, can run commands that are prefixed with the $ prompt.

 # prompt The `root` user must run commands that are prefixed with the # prompt. You can also prefix these commands with the **sudo** command, if available, to run them.

Document change history

This version of the guide replaces and obsoletes all earlier versions.

The following table describes the most recent changes:

Revision Date	Summary of Changes
October 15, 2014	• Updates for Juno: updated all configuration tables, include sample configuration files, add chapter for Data processing service, update and enhance driver configuration.
April 16, 2014	• Update for Icehouse: Updated all configuration tables, include sample configuration files, add chapters for Database service, Orchestration, and Telemetry.
March 11, 2014	• Sorted component listing. Moved procedures to the *Cloud Administrator Guide*
January 9, 2014	• Removes content addressed in installation, merges duplicated content, and revises legacy references.
October 17, 2013	• Havana release.
August 16, 2013	• Moves Block Storage driver configuration information from the *Block Storage Administration Guide* to this reference.
June 10, 2013	• 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 types 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 A floating point number like `0.25` or `1000`.

```
# Sleep time in seconds for polling an ongoing async
 task
# (floating point value)
task_poll_interval = 0.5
```

integer value An integer number is a number without fractional components, like `0` or `42`.

```
# The port which the OpenStack Compute service
 listens on.
# (integer value)
compute_port = 8774
```

list value Represents values of other types, separated by commas. As an example, the following sets `allowed_rpc_exception_modules` to a list containing the four elements `oslo.messaging.exceptions`, `nova.exception`, `cinder.exception`, and `exceptions`:

```
# 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
```

multi valued A multi-valued option is a string value and can be given more than once, all values will be used.

```
# Driver or drivers to handle sending notifications.
 (multi
# valued)
notification_driver = nova.openstack.common.
notifier.rpc_notifier
notification_driver = ceilometer.compute.
nova_notifier
```

string value Strings can be optionally enclosed with single or double quotes.

```
# 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] "
```

Sections

Configuration options are grouped by section. Most configuration files support 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_passsword='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. Bare metal

The Bare metal service is capable of managing and provisioning physical machines. The configuration file of this module is `/etc/ironic/ironic.conf`.

The following tables provide a comprehensive list of the Bare metal service configuration options.

Table 1.1. Description of agent configuration options

Configuration option = Default value	Description
[agent]	
`agent_api_version` = *v1*	(StrOpt) API version to use for communicating with the ramdisk agent.
`agent_erase_devices_priority` = *None*	(IntOpt) Priority to run in-band erase devices via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 10 for the GenericHardwareManager). If set to 0, will not run during cleaning.
`agent_pxe_append_params` = *nofb nomodeset vga=normal*	(StrOpt) Additional append parameters for baremetal PXE boot.
`agent_pxe_bootfile_name` = *pxelinux.0*	(StrOpt) Neutron bootfile DHCP parameter.
`agent_pxe_config_template` = *$pybasedir/drivers/modules/agent_config.template*	(StrOpt) Template file for PXE configuration.
`heartbeat_timeout` = *300*	(IntOpt) Maximum interval (in seconds) for agent heartbeats.
`manage_tftp` = *True*	(BoolOpt) Whether Ironic will manage TFTP files for the deploy ramdisks. If set to False, you will need to configure your own TFTP server that allows booting the deploy ramdisks.

Table 1.2. Description of AMQP configuration options

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.
`notification_driver` = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
`notification_topics` = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
`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 1.3. Description of AMT configuration options

Configuration option = Default value	Description
[amt]	
`action_wait` = *10*	(IntOpt) Amount of time (in seconds) to wait, before retrying an AMT operation
`max_attempts` = *3*	(IntOpt) Maximum number of times to attempt an AMT operation, before failing
`protocol` = *http*	(StrOpt) Protocol used for AMT endpoint, support http/https

Table 1.4. Description of API configuration options

Configuration option = Default value	Description
[api]	
host_ip = *0.0.0.0*	(StrOpt) The IP address on which ironic-api listens.
max_limit = *1000*	(IntOpt) The maximum number of items returned in a single response from a collection resource.
port = *6385*	(IntOpt) The TCP port on which ironic-api listens.

Table 1.5. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.

Configuration option = Default value	Description
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
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 = *10*	(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.

Configuration option = Default value	Description
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 1.6. Description of authorization configuration options

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = *keystone*	(StrOpt) Authentication strategy used by ironic-api: one of "keystone" or "noauth". "noauth" should not be used in a production environment because all authentication will be disabled.

Table 1.7. Description of Cisco UCS configuration options

Configuration option = Default value	Description
[cisco_ucs]	
action_interval = *5*	(IntOpt) Amount of time in seconds to wait in between power operations
max_retry = *6*	(IntOpt) Number of times a power operation needs to be retried

Table 1.8. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
bindir = */usr/local/bin*	(StrOpt) Directory where ironic binaries are installed.
enabled_drivers = *pxe_ipmitool*	(ListOpt) Specify the list of drivers to load during service initialization. Missing drivers, or drivers which fail to initialize, will prevent the conductor service from starting. The option default is a recommended set of production-oriented drivers. A complete list of drivers present on your system may be found by enumerating the "ironic.drivers" entrypoint. An example may be found in the developer documentation online.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
force_raw_images = *True*	(BoolOpt) If True, convert backing images to "raw" disk image format.
grub_config_template = *$pybasedir/common/grub_conf.template*	(StrOpt) Template file for grub configuration file.
hash_distribution_replicas = *1*	(IntOpt) [Experimental Feature] Number of hosts to map onto each hash partition. Setting this to more than one will cause additional conductor services to prepare deployment environments and potentially allow the Ironic cluster to recover more quickly if a conductor instance is terminated.
hash_partition_exponent = *5*	(IntOpt) Exponent to determine number of hash partitions to use when distributing load across conductors. Larger values will result in more even distribution of load and less load when rebalancing the ring, but more memory usage. Number of partitions per conductor is (2^hash_partition_exponent). This determines the granularity of rebalancing: given 10 hosts, and an exponent of

Configuration option = Default value	Description
	the 2, there are 40 partitions in the ring. A few thousand partitions should make rebalancing smooth in most cases. The default is suitable for up to a few hundred conductors. Too many partitions has a CPU impact.
host = *localhost*	(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.
isolinux_bin = */usr/lib/syslinux/isolinux.bin*	(StrOpt) Path to isolinux binary file.
isolinux_config_template = *$pybasedir/common/isolinux_config.template*	(StrOpt) Template file for isolinux configuration file.
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
my_ip = *10.0.0.1*	(StrOpt) IP address of this host. If unset, will determine the IP programmatically. If unable to do so, will use "127.0.0.1".
parallel_image_downloads = *False*	(BoolOpt) Run image downloads and raw format conversions in parallel.
periodic_interval = *60*	(IntOpt) Seconds between running periodic tasks.
pybasedir = */usr/lib/python/site-packages/ironic/ironic*	(StrOpt) Directory where the ironic python module is installed.
rootwrap_config = */etc/ironic/rootwrap.conf*	(StrOpt) Path to the rootwrap configuration file to use for running commands as root.
run_external_periodic_tasks = *True*	(BoolOpt) Some periodic tasks can be run in a separate process. Should we run them here?
state_path = *$pybasedir*	(StrOpt) Top-level directory for maintaining ironic's state.
tempdir = *None*	(StrOpt) Explicitly specify the temporary working directory.

Table 1.9. Description of conductor configuration options

Configuration option = Default value	Description
[conductor]	
api_url = *None*	(StrOpt) URL of Ironic API service. If not set ironic can get the current value from the keystone service catalog.
check_provision_state_interval = *60*	(IntOpt) Interval between checks of provision timeouts, in seconds.
clean_nodes = *True*	(BoolOpt) Cleaning is a configurable set of steps, such as erasing disk drives, that are performed on the node to ensure it is in a baseline state and ready to be deployed to. This is done after instance deletion, and during the transition from a "managed" to "available" state. When enabled, the particular steps performed to clean a node depend on which driver that node is managed by; see the individual driver's documentation for details. NOTE: The introduction of the cleaning operation causes instance deletion to take significantly longer. In an environment where all tenants are trusted (eg, because there is only one tenant), this option could be safely disabled.
configdrive_swift_container = *ironic_configdrive_container*	(StrOpt) Name of the Swift container to store config drive data. Used when configdrive_use_swift is True.
configdrive_use_swift = *False*	(BoolOpt) Whether to upload the config drive to Swift.
deploy_callback_timeout = *1800*	(IntOpt) Timeout (seconds) to wait for a callback from a deploy ramdisk. Set to 0 to disable timeout.
force_power_state_during_sync = *True*	(BoolOpt) During sync_power_state, should the hardware power state be set to the state recorded in the database

Configuration option = Default value	Description
	(True) or should the database be updated based on the hardware state (False).
heartbeat_interval = 10	(IntOpt) Seconds between conductor heart beats.
heartbeat_timeout = 60	(IntOpt) Maximum time (in seconds) since the last check-in of a conductor. A conductor is considered inactive when this time has been exceeded.
inspect_timeout = 1800	(IntOpt) Timeout (seconds) for waiting for node inspection. 0 - unlimited.
node_locked_retry_attempts = 3	(IntOpt) Number of attempts to grab a node lock.
node_locked_retry_interval = 1	(IntOpt) Seconds to sleep between node lock attempts.
periodic_max_workers = 8	(IntOpt) Maximum number of worker threads that can be started simultaneously by a periodic task. Should be less than RPC thread pool size.
power_state_sync_max_retries = 3	(IntOpt) During sync_power_state failures, limit the number of times Ironic should try syncing the hardware node power state with the node power state in DB
send_sensor_data = False	(BoolOpt) Enable sending sensor data message via the notification bus
send_sensor_data_interval = 600	(IntOpt) Seconds between conductor sending sensor data message to ceilometer via the notification bus.
send_sensor_data_types = ALL	(ListOpt) List of comma separated meter types which need to be sent to Ceilometer. The default value, "ALL", is a special value meaning send all the sensor data.
sync_local_state_interval = 180	(IntOpt) When conductors join or leave the cluster, existing conductors may need to update any persistent local state as nodes are moved around the cluster. This option controls how often, in seconds, each conductor will check for nodes that it should "take over". Set it to a negative value to disable the check entirely.
sync_power_state_interval = 60	(IntOpt) Interval between syncing the node power state to the database, in seconds.
workers_pool_size = 100	(IntOpt) The size of the workers greenthread pool.

Table 1.10. Description of console configuration options

Configuration option = Default value	Description
[console]	
subprocess_checking_interval = 1	(IntOpt) Time interval (in seconds) for checking the status of console subprocess.
subprocess_timeout = 10	(IntOpt) Time (in seconds) to wait for the console subprocess to start.
terminal = shellinaboxd	(StrOpt) Path to serial console terminal program
terminal_cert_dir = None	(StrOpt) Directory containing the terminal SSL cert(PEM) for serial console access
terminal_pid_dir = None	(StrOpt) Directory for holding terminal pid files. If not specified, the temporary directory will be used.

Table 1.11. Description of database configuration options

Configuration option = Default value	Description
[database]	
backend = sqlalchemy	(StrOpt) The back end to use for the database.
connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the database.

Configuration option = Default value	Description
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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = 10	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify an infinite retry count.
min_pool_size = 1	(IntOpt) Minimum number of SQL connections to keep open in a pool.
mysql_engine = InnoDB	(StrOpt) MySQL engine to use.
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.
slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = oslo.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 1.12. Description of logging configuration options

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.
pecan_debug = False	(BoolOpt) Enable pecan debug mode. WARNING: this is insecure and should not be used in a production environment.

Table 1.13. Description of deploy configuration options

Configuration option = Default value	Description
[deploy]	
dd_block_size = *1M*	(StrOpt) Block size to use when writing to the nodes disk.
efi_system_partition_size = *200*	(IntOpt) Size of EFI system partition in MiB when configuring UEFI systems for local boot.
iscsi_verify_attempts = *3*	(IntOpt) Maximum attempts to verify an iSCSI connection is active, sleeping 1 second between attempts.

Table 1.14. Description of DHCP configuration options

Configuration option = Default value	Description
[dhcp]	
dhcp_provider = *neutron*	(StrOpt) DHCP provider to use. "neutron" uses Neutron, and "none" uses a no-op provider.

Table 1.15. Description of disk partitioner configuration options

Configuration option = Default value	Description
[disk_partitioner]	
check_device_interval = *1*	(IntOpt) After Ironic has completed creating the partition table, it continues to check for activity on the attached iSCSI device status at this interval prior to copying the image to the node, in seconds
check_device_max_retries = *20*	(IntOpt) The maximum number of times to check that the device is not accessed by another process. If the device is still busy after that, the disk partitioning will be treated as having failed.

Table 1.16. Description of DRAC configuration options

Configuration option = Default value	Description
[drac]	
client_retry_count = *5*	(IntOpt) In case there is a communication failure, the DRAC client is going to resend the request as many times as defined in this setting.
client_retry_delay = *5*	(IntOpt) In case there is a communication failure, the DRAC client is going to wait for as many seconds as defined in this setting before resending the request.

Table 1.17. Description of glance configuration options

Configuration option = Default value	Description
[glance]	
allowed_direct_url_schemes =	(ListOpt) A list of URL schemes that can be downloaded directly via the direct_url. Currently supported schemes: [file].
auth_strategy = *keystone*	(StrOpt) Authentication strategy to use when connecting to glance. Only "keystone" and "noauth" are currently supported by ironic.
glance_api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL (https) requests to glance.
glance_api_servers = *None*	(ListOpt) A list of the glance api servers available to ironic. Prefix with https:// for SSL-based glance API servers. Format is [hostname|IP]:port.

Configuration option = Default value	Description
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.
swift_account = *None*	(StrOpt) The account that Glance uses to communicate with Swift. The format is "AUTH_uuid". "uuid" is the UUID for the account configured in the glance-api.conf. Required for temporary URLs. For example: "AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30". Swift temporary URL format: "endpoint_url/api_version/account/container/object_id"
swift_api_version = *v1*	(StrOpt) The Swift API version to create a temporary URL for. Defaults to "v1". Swift temporary URL format: "endpoint_url/api_version/account/container/object_id"
swift_container = *glance*	(StrOpt) The Swift container Glance is configured to store its images in. Defaults to "glance", which is the default in glance-api.conf. Swift temporary URL format: "endpoint_url/api_version/account/container/object_id"
swift_endpoint_url = *None*	(StrOpt) The "endpoint" (scheme, hostname, optional port) for the Swift URL of the form "endpoint_url/api_version/account/container/object_id". Do not include trailing "/". For example, use "https://swift.example.com". Required for temporary URLs.
swift_store_multiple_containers_seed = *0*	(IntOpt) This should match a config by the same name in the Glance configuration file. When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.
swift_temp_url_duration = *1200*	(IntOpt) The length of time in seconds that the temporary URL will be valid for. Defaults to 20 minutes. If some deploys get a 401 response code when trying to download from the temporary URL, try raising this duration.
swift_temp_url_key = *None*	(StrOpt) The secret token given to Swift to allow temporary URL downloads. Required for temporary URLs.

Table 1.18. Description of iLO configuration options

Configuration option = Default value	Description
[ilo]	
clean_priority_clear_secure_boot_keys = *0*	(IntOpt) Priority for clear_secure_boot_keys clean step. This step is not enabled by default. It can be enabled to to clear all secure boot keys enrolled with iLO.
clean_priority_erase_devices = *None*	(IntOpt) Priority for erase devices clean step. If unset, it defaults to 10. If set to 0, the step will be disabled and will not run during cleaning.
clean_priority_reset_bios_to_default = *10*	(IntOpt) Priority for reset_bios_to_default clean step.
clean_priority_reset_ilo = *1*	(IntOpt) Priority for reset_ilo clean step.
clean_priority_reset_ilo_credential = *30*	(IntOpt) Priority for reset_ilo_credential clean step. This step requires "ilo_change_password" parameter to be updated in nodes's driver_info with the new password.
clean_priority_reset_secure_boot_keys_to_default = *20*	(IntOpt) Priority for reset_secure_boot_keys clean step. This step will reset the secure boot keys to manufacturing defaults.

Configuration option = Default value	Description
client_port = 443	(IntOpt) Port to be used for iLO operations
client_timeout = 60	(IntOpt) Timeout (in seconds) for iLO operations
power_retry = 6	(IntOpt) Number of times a power operation needs to be retried
power_wait = 2	(IntOpt) Amount of time in seconds to wait in between power operations
swift_ilo_container = ironic_ilo_container	(StrOpt) The Swift iLO container to store data.
swift_object_expiry_timeout = 900	(IntOpt) Amount of time in seconds for Swift objects to auto-expire.

Table 1.19. Description of inspector configuration options

Configuration option = Default value	Description
[inspector]	
enabled = False	(BoolOpt) whether to enable inspection using ironic-inspector
service_url = None	(StrOpt) ironic-inspector HTTP endpoint. If this is not set, the ironic-inspector client default (http://127.0.0.1:5050) will be used.
status_check_period = 60	(IntOpt) period (in seconds) to check status of nodes on inspection

Table 1.20. Description of IPMI configuration options

Configuration option = Default value	Description
[ipmi]	
min_command_interval = 5	(IntOpt) Minimum time, in seconds, between IPMI operations sent to a server. There is a risk with some hardware that setting this too low may cause the BMC to crash. Recommended setting is 5 seconds.
retry_timeout = 60	(IntOpt) Maximum time in seconds to retry IPMI operations. There is a tradeoff when setting this value. Setting this too low may cause older BMCs to crash and require a hard reset. However, setting too high can cause the sync power state periodic task to hang when there are slow or unresponsive BMCs.

Table 1.21. Description of iRMC configuration options

Configuration option = Default value	Description
[irmc]	
auth_method = basic	(StrOpt) Authentication method to be used for iRMC operations, either "basic" or "digest"
client_timeout = 60	(IntOpt) Timeout (in seconds) for iRMC operations
port = 443	(IntOpt) Port to be used for iRMC operations, either 80 or 443
sensor_method = ipmitool	(StrOpt) Sensor data retrieval method, either "ipmitool" or "scci"

Table 1.22. Description of keystone configuration options

Configuration option = Default value	Description
[keystone]	

Configuration option = Default value	Description
region_name = *None*	(StrOpt) The region used for getting endpoints of OpenStackservices.

Table 1.23. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_exception_format_errors = *False*	(BoolOpt) Used if there is a formatting error when generating an exception message (a programming error). If True, raise an exception; if False, use the unformatted message.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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 ERROR %(name)s %(instance)s*	(StrOpt) Prefix each line of exception output with this format.
publish_errors = *False*	(BoolOpt) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.

Configuration option = Default value	Description
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 changed in J to honor RFC5424.
use_syslog_rfc_format = True	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in K, and will be removed in M, along with this option.
verbose = False	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 1.24. Description of neutron configuration options

Configuration option = Default value	Description
[neutron]	
auth_strategy = keystone	(StrOpt) Default authentication strategy to use when connecting to neutron. Can be either "keystone" or "noauth". Running neutron in noauth mode (related to but not affected by this setting) is insecure and should only be used for testing.
cleaning_network_uuid = None	(StrOpt) UUID of the network to create Neutron ports on when booting to a ramdisk for cleaning/zapping using Neutron DHCP
retries = 3	(IntOpt) Client retries in the case of a failed request.
url = http://$my_ip:9696	(StrOpt) URL for connecting to neutron.
url_timeout = 30	(IntOpt) Timeout value for connecting to neutron in seconds.

Table 1.25. Description of policy configuration options

Configuration option = Default value	Description
[oslo_policy]	
policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.json	(StrOpt) The JSON file that defines policies.

Table 1.26. Description of PXE configuration options

Configuration option = Default value	Description
[pxe]	
default_ephemeral_format = ext4	(StrOpt) Default file system format for ephemeral partition, if one is created.
disk_devices = cciss/c0d0,sda,hda,vda	(StrOpt) The disk devices to scan while doing the deploy.
http_root = /httpboot	(StrOpt) ironic-conductor node's HTTP root path.
http_url = None	(StrOpt) ironic-conductor node's HTTP server URL. Example: http://192.1.2.3:8080

Configuration option = Default value	Description
image_cache_size = *20480*	(IntOpt) Maximum size (in MiB) of cache for master images, including those in use.
image_cache_ttl = *10080*	(IntOpt) Maximum TTL (in minutes) for old master images in cache.
images_path = */var/lib/ironic/images/*	(StrOpt) On the ironic-conductor node, directory where images are stored on disk.
instance_master_path = */var/lib/ironic/master_images*	(StrOpt) On the ironic-conductor node, directory where master instance images are stored on disk.
ipxe_boot_script = *$pybasedir/drivers/modules/boot.ipxe*	(StrOpt) On ironic-conductor node, the path to the main iPXE script file.
ipxe_enabled = *False*	(BoolOpt) Enable iPXE boot.
pxe_append_params = *nofb nomodeset vga=normal*	(StrOpt) Additional append parameters for baremetal PXE boot.
pxe_bootfile_name = *pxelinux.0*	(StrOpt) Bootfile DHCP parameter.
pxe_config_template = *$pybasedir/drivers/modules/pxe_config.template*	(StrOpt) On ironic-conductor node, template file for PXE configuration.
tftp_master_path = */tftpboot/master_images*	(StrOpt) On ironic-conductor node, directory where master TFTP images are stored on disk.
tftp_root = */tftpboot*	(StrOpt) ironic-conductor node's TFTP root path.
tftp_server = *$my_ip*	(StrOpt) IP address of ironic-conductor node's TFTP server.
uefi_pxe_bootfile_name = *elilo.efi*	(StrOpt) Bootfile DHCP parameter for UEFI boot mode.
uefi_pxe_config_template = *$pybasedir/drivers/modules/elilo_efi_pxe_config.template*	(StrOpt) On ironic-conductor node, template file for PXE configuration for UEFI boot loader.

Table 1.27. Description of Redis configuration options

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 1.28. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user run-

Configuration option = Default value	Description
	ning the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 1.29. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.

Configuration option = Default value	Description
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_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

Table 1.30. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
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_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

Table 1.31. Description of SeaMicro configuration options

Configuration option = Default value	Description
[seamicro]	
action_timeout = 10	(IntOpt) Seconds to wait for power action to be completed
max_retry = 3	(IntOpt) Maximum retries for SeaMicro operations

Table 1.32. Description of SNMP configuration options

Configuration option = Default value	Description
[snmp]	
power_timeout = *10*	(IntOpt) Seconds to wait for power action to be completed

Table 1.33. Description of SSH configuration options

Configuration option = Default value	Description
[ssh]	
libvirt_uri = *qemu:///system*	(StrOpt) libvirt URI

Table 1.34. Description of swift configuration options

Configuration option = Default value	Description
[swift]	
swift_max_retries = *2*	(IntOpt) Maximum number of times to retry a Swift request, before failing.

Table 1.35. Description of VirtualBox configuration options

Configuration option = Default value	Description
[virtualbox]	
port = *18083*	(IntOpt) Port on which VirtualBox web service is listening.

Table 1.36. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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.

2. Block Storage

Table of Contents

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 Storage (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 backend 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 user name 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).

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://git.openstack.org/cgit/openstack/cinder/). To set a volume driver, use the `volume_driver` flag. The default is:

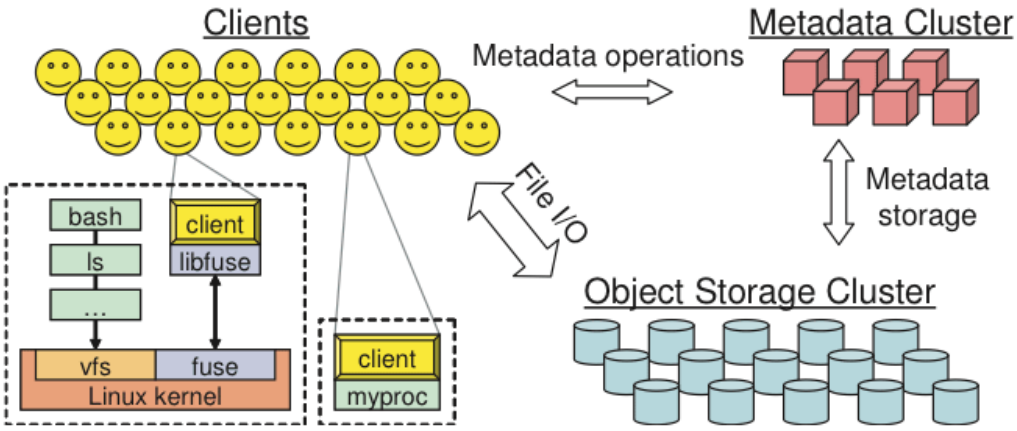
```
volume_driver = cinder.volume.drivers.lvm.LVMISCSIDriver
```

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 2.1. 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.

Deprecation notice

The `volume_tmp_dir` option has been deprecated and replaced by `image_conversion_dir`.

Table 2.1. Description of Ceph storage configuration options

Configuration option = Default value	Description
[DEFAULT]	
`rados_connect_timeout` = *-1*	(IntOpt) Timeout value (in seconds) used when connecting to ceph cluster. If value < 0, no timeout is set and default librados value is used.
`rbd_ceph_conf` =	(StrOpt) Path to the ceph configuration file
`rbd_flatten_volume_from_snapshot` = *False*	(BoolOpt) Flatten volumes created from snapshots to remove dependency from volume to snapshot

Configuration option = Default value	Description
rbd_max_clone_depth = 5	(IntOpt) Maximum number of nested volume clones that are taken before a flatten occurs. Set to 0 to disable cloning.
rbd_pool = rbd	(StrOpt) The RADOS pool where rbd volumes are stored
rbd_secret_uuid = None	(StrOpt) The libvirt uuid of the secret for the rbd_user volumes
rbd_store_chunk_size = 4	(IntOpt) Volumes will be chunked into objects of this size (in megabytes).
rbd_user = None	(StrOpt) The RADOS client name for accessing rbd volumes - only set when using cephx authentication
volume_tmp_dir = None	(StrOpt) Directory where temporary image files are stored when the volume driver does not write them directly to the volume. Warning: this option is now deprecated, please use image_conversion_dir instead.

Dell EqualLogic volume driver

The Dell EqualLogic volume driver interacts with configured EqualLogic arrays and supports various operations.

Supported operations

- Create, delete, attach, and detach volumes.
- Create, list, and delete volume snapshots.
- Clone a volume.

The OpenStack Block Storage service supports:

- Multiple instances of Dell EqualLogic Groups or Dell EqualLogic Group Storage Pools and multiple pools on a single array.
- Multiple instances of Dell EqualLogic Groups or Dell EqualLogic Group Storage Pools or multiple pools on a single array.

The Dell EqualLogic volume driver's ability to access the EqualLogic Group is dependent upon the generic block storage driver's SSH settings in the `/etc/cinder/cinder.conf` file (see the section called “Block Storage sample configuration files” [123] for reference).

Table 2.2. Description of Dell EqualLogic volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
eqlx_chap_login = admin	(StrOpt) Existing CHAP account name. Note that this option is deprecated in favour of "chap_username" as specified in cinder/volume/driver.py and will be removed in next release.
eqlx_chap_password = password	(StrOpt) Password for specified CHAP account name. Note that this option is deprecated in favour of "chap_password" as specified in cinder/volume/driver.py and will be removed in the next release
eqlx_cli_max_retries = 5	(IntOpt) Maximum retry count for reconnection. Default is 5.

Configuration option = Default value	Description
eqlx_cli_timeout = *30*	(IntOpt) Timeout for the Group Manager cli command execution. Default is 30.
eqlx_group_name = *group-0*	(StrOpt) Group name to use for creating volumes. Defaults to "group-0".
eqlx_pool = *default*	(StrOpt) Pool in which volumes will be created. Defaults to "default".
eqlx_use_chap = *False*	(BoolOpt) Use CHAP authentication for targets. Note that this option is deprecated in favour of "use_chap_auth" as specified in cinder/volume/driver.py and will be removed in next release.

The following sample `/etc/cinder/cinder.conf` configuration lists the relevant settings for a typical Block Storage service using a single Dell EqualLogic Group:

Example 2.1. Default (single-instance) configuration

```
[DEFAULT]
#Required settings

volume_driver = cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
san_ip = IP_EQLX
san_login = SAN_UNAME
san_password = SAN_PW
eqlx_group_name = EQLX_GROUP
eqlx_pool = EQLX_POOL

#Optional settings

san_thin_provision = true|false
eqlx_use_chap = true|false
eqlx_chap_login = EQLX_UNAME
eqlx_chap_password = EQLX_PW
eqlx_cli_timeout = 30
eqlx_cli_max_retries = 5
san_ssh_port = 22
ssh_conn_timeout = 30
san_private_key = SAN_KEY_PATH
ssh_min_pool_conn = 1
ssh_max_pool_conn = 5
```

In this example, replace the following variables accordingly:

IP_EQLX The IP address used to reach the Dell EqualLogic Group through SSH. This field has no default value.

SAN_UNAME The user name to login to the Group manager via SSH at the `san_ip`. Default user name is `grpadmin`.

SAN_PW The corresponding password of *SAN_UNAME*. Not used when `san_private_key` is set. Default password is `password`.

EQLX_GROUP The group to be used for a pool where the Block Storage service will create volumes and snapshots. Default group is `group-0`.

EQLX_POOL	The pool where the Block Storage service will create volumes and snapshots. Default pool is `default`. This option cannot be used for multiple pools utilized by the Block Storage service on a single Dell EqualLogic Group.
EQLX_UNAME	The CHAP login account for each volume in a pool, if `eqlx_use_chap` is set to `true`. Default account name is `chapadmin`.
EQLX_PW	The corresponding password of *EQLX_UNAME*. The default password is randomly generated in hexadecimal, so you must set this password manually.
SAN_KEY_PATH (optional)	The filename of the private key used for SSH authentication. This provides password-less login to the EqualLogic Group. Not used when `san_password` is set. There is no default value.

In addition, enable thin provisioning for SAN volumes using the default `san_thin_provision = true` setting.

Example 2.2. Multi back-end Dell EqualLogic configuration

The following example shows the typical configuration for a Block Storage service that uses two Dell EqualLogic back ends:

```
enabled_backends = backend1,backend2
san_ssh_port = 22
ssh_conn_timeout = 30
san_thin_provision = true

[backend1]
volume_driver = cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
volume_backend_name = backend1
san_ip = IP_EQLX1
san_login = SAN_UNAME
san_password = SAN_PW
eqlx_group_name = EQLX_GROUP
eqlx_pool = EQLX_POOL

[backend2]
volume_driver = cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
volume_backend_name = backend2
san_ip = IP_EQLX2
san_login = SAN_UNAME
san_password = SAN_PW
eqlx_group_name = EQLX_GROUP
eqlx_pool = EQLX_POOL
```

In this example:

- Thin provisioning for SAN volumes is enabled (`san_thin_provision = true`). This is recommended when setting up Dell EqualLogic back ends.
- Each Dell EqualLogic back-end configuration (`[backend1]` and `[backend2]`) has the same required settings as a single back-end configuration, with the addition of `volume_backend_name`.

- The `san_ssh_port` option is set to its default value, 22. This option sets the port used for SSH.
- The `ssh_conn_timeout` option is also set to its default value, 30. This option sets the timeout in seconds for CLI commands over SSH.
- The `IP_EQLX1` and `IP_EQLX2` refer to the IP addresses used to reach the Dell EqualLogic Group of `backend1` and `backend2` through SSH, respectively.

For information on configuring multiple back ends, see Configure a multiple-storage back end.

Dell Storage Center Fibre Channel and iSCSI drivers

The Dell Storage Center volume driver interacts with configured Storage Center arrays.

The Dell Storage Center driver manages Storage Center arrays through Enterprise Manager. Enterprise Manager connection settings and Storage Center options are defined in the `cinder.conf` file.

Prerequisite: Dell Enterprise Manager 2015 R1 or later must be used.

Supported operations

The Dell Storage Center volume driver provides the following Cinder volume operations:

- Create, delete, attach (map), and detach (unmap) 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.
- Extend a volume.

iSCSI configuration

Use the following instructions to update the configuration file for iSCSI:

Example 2.3. Sample iSCSI Configuration

```
default_volume_type = delliscsi
enabled_backends = delliscsi

[delliscsi]
# Name to give this storage backend
volume_backend_name = delliscsi
# The iSCSI driver to load
volume_driver = cinder.volume.drivers.dell.dell_storagecenter_iscsi.
DellStorageCenterISCSIDriver
```

```
# IP address of Enterprise Manager
san_ip = 172.23.8.101
# Enterprise Manager user name
san_login = Admin
# Enterprise Manager password
san_password = secret
# The Storage Center iSCSI IP address
iscsi_ip_address = 192.168.0.20
# The Storage Center serial number to use
dell_sc_ssn = 64702

# ==Optional settings==
# The Enterprise Manager API port
dell_sc_api_port = 3033
# Server folder to place new server definitions
dell_sc_server_folder = devstacksrv
# Volume folder to place created volumes
dell_sc_volume_folder = devstackvol/Cinder
# The iSCSI IP port
iscsi_port = 3260
```

Fibre Channel configuration

Use the following instructions to update the configuration file for fibre channel:

Example 2.4. Sample FC configuration

```
default_volume_type = dellfc
enabled_backends = dellfc

[dellfc]
# Name to give this storage backend
volume_backend_name = dellfc
# The FC driver to load
volume_driver = cinder.volume.drivers.dell.dell_storagecenter_fc.
DellStorageCenterFCDriver
# IP address of Enterprise Manager
san_ip = 172.23.8.101
# Enterprise Manager user name
san_login = Admin
# Enterprise Manager password
san_password = secret
# The Storage Center serial number to use
dell_sc_ssn = 64702

# Optional settings

# The Enterprise Manager API port
dell_sc_api_port = 3033
# Server folder to place new server definitions
dell_sc_server_folder = devstacksrv
# Volume folder to place created volumes
dell_sc_volume_folder = devstackvol/Cinder
```

Driver options

The following table contains the configation options specific to the Dell Storage Center volume driver.

Table 2.3. Description of Dell Storage Center volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`dell_sc_api_port = 3033`	(IntOpt) Dell API port
`dell_sc_server_folder = openstack`	(StrOpt) Name of the server folder to use on the Storage Center
`dell_sc_ssn = 64702`	(IntOpt) Storage Center System Serial Number
`dell_sc_volume_folder = openstack`	(StrOpt) Name of the volume folder to use on the Storage Center

EMC VMAX iSCSI and FC drivers

The EMC VMAX drivers, `EMCVMAXISCSIDriver` and `EMCVMAXFCDriver`, support the use of EMC VMAX storage arrays under OpenStack Block Storage. They both provide equivalent functions and differ only in support for their respective host attachment methods.

The drivers perform volume operations by communicating with the backend VMAX 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 VMAX storage operations.

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

System requirements

EMC SMI-S Provider V4.6.2.8 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 is supported.

Supported operations

VMAX drivers support 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.
- Clone a volume.
- Extend a volume.

- Retype a volume.
- Create a volume from a snapshot.

VMAX drivers also support the following features:

- FAST automated storage tiering policy.
- Dynamic masking view creation.
- Striped volume creation.

Set up the VMAX drivers

Procedure 2.1. To set up the EMC VMAX drivers

1. Install the python-pywbem package for your distribution. See the section called "Install the python-pywbem package" [27].
2. Download SMI-S from PowerLink and install it. Add your VMAX arrays to SMI-S.

   For information, see the section called "Set up SMI-S" [27] and the SMI-S release notes.
3. Change configuration files. See the section called "`cinder.conf` configuration file" [28] and the section called "`cinder_emc_config_CONF_GROUP_ISCSI.xml` configuration file" [28].
4. Configure connectivity. For FC driver, see the section called "FC Zoning with VMAX" [29]. For iSCSI driver, see the section called "iSCSI with VMAX" [29].

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 VMAX drivers. 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 VMAX drivers.

`cinder.conf` configuration file

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

Add the following entries, where `10.10.61.45` is the IP address of the VMAX iSCSI target:

```
enabled_backends = CONF_GROUP_ISCSI, CONF_GROUP_FC
[CONF_GROUP_ISCSI]
iscsi_ip_address = 10.10.61.45
volume_driver = cinder.volume.drivers.emc.emc_vmax_iscsi.EMCVMAXISCSIDriver
cinder_emc_config_file = /etc/cinder/cinder_emc_config_CONF_GROUP_ISCSI.xml
volume_backend_name=ISCSI_backend
[CONF_GROUP_FC]
volume_driver = cinder.volume.drivers.emc.emc_vmax_fc.EMCVMAXFCDriver
cinder_emc_config_file = /etc/cinder/cinder_emc_config_CONF_GROUP_FC.xml
volume_backend_name=FC_backend
```

In this example, two backend configuration groups are enabled: `CONF_GROUP_ISCSI` and `CONF_GROUP_FC`. Each configuration group has a section describing unique parameters for connections, drivers, the volume_backend_name, and the name of the EMC-specific configuration file containing additional settings. Note that the file name is in the format `/etc/cinder/cinder_emc_config_[confGroup].xml`.

Once the `cinder.conf` and EMC-specific configuration files have been created, cinder commands need to be issued in order to create and associate OpenStack volume types with the declared volume_backend_names:

```
$ cinder type-create VMAX_ISCSI
$ cinder type-key VMAX_ISCSI set volume_backend_name=ISCSI_backend
$ cinder type-create VMAX_FC
$ cinder type-key VMAX_FC set volume_backend_name=FC_backend
```

By issuing these commands, the Block Storage volume type `VMAX_ISCSI` is associated with the ISCSI_backend, and the type VMAX_FC is associated with the FC_backend.

Restart the `cinder-volume` service.

`cinder_emc_config_CONF_GROUP_ISCSI.xml` configuration file

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

Add the following lines to the XML file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<EMC>
    <EcomServerIp>1.1.1.1</EcomServerIp>
    <EcomServerPort>00</EcomServerPort>
    <EcomUserName>user1</EcomUserName>
    <EcomPassword>password1</EcomPassword>
    <PortGroups>
      <PortGroup>OS-PORTGROUP1-PG</PortGroup>
      <PortGroup>OS-PORTGROUP2-PG</PortGroup>
    </PortGroups>
   <Array>111111111111</Array>
   <Pool>FC_GOLD1</Pool>
   <FastPolicy>GOLD1</FastPolicy>
</EMC>
```

Where:

- `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.
- `PortGroups` supplies the names of VMAX port groups that have been pre-configured to expose volumes managed by this backend. Each supplied port group should have sufficient number and distribution of ports (across directors and switches) as to ensure adequate bandwidth and failure protection for the volume connections. `PortGroups` can contain one or more port groups of either iSCSI or FC ports. When a dynamic masking view is created by the VMAX driver, the port group is chosen randomly from the `PortGroup` list, to evenly distribute load across the set of groups provided. Make sure that the `PortGroups` set contains either all FC or all iSCSI port groups (for a given backend), as appropriate for the configured driver (iSCSI or FC).
- The `Array` tag holds the unique VMAX array serial number.
- The `Pool` tag holds the unique pool name within a given array. For backends not using FAST automated tiering, the pool is a single pool that has been created by the administrator. For backends exposing FAST policy automated tiering, the pool is the bind pool to be used with the FAST policy.
- The `FastPolicy` tag conveys the name of the FAST Policy to be used. By including this tag, volumes managed by this backend are treated as under FAST control. Omitting the `FastPolicy` tag means FAST is not enabled on the provided storage pool.

FC Zoning with VMAX

Zone Manager is recommended when using the VMAX FC driver, especially for larger configurations where pre-zoning would be too complex and open-zoning would raise security concerns.

iSCSI with VMAX

- Make sure the iscsi-initiator-utils package is installed on the host (use apt-get, zypper, or yum, depending on Linux flavor).
- Verify host is able to ping VMAX iSCSI target ports.

VMAX masking view and group naming info

Masking view names

Masking views are dynamically created by the VMAX FC and iSCSI drivers using the following naming conventions:

```
OS-[shortHostName][poolName]-I-MV (for Masking Views using iSCSI)
```

```
OS-[shortHostName][poolName]-F-MV (for Masking Views using FC)
```

Initiator group names

For each host that is attached to VMAX volumes using the drivers, an initiator group is created or re-used (per attachment type). All initiators of the appropriate type known for that host are included in the group. At each new attach volume operation, the VMAX driver retrieves the initiators (either WWNNs or IQNs) from OpenStack and adds or updates the contents of the Initiator Group as required. Names are of the following format:

```
OS-[shortHostName]-I-IG (for iSCSI initiators)
```

```
OS-[shortHostName]-F-IG (for Fibre Channel initiators)
```

Note

Hosts attaching to VMAX storage managed by the OpenStack environment cannot also be attached to storage on the same VMAX not being managed by OpenStack. This is due to limitations on VMAX Initiator Group membership.

FA port groups

VMAX array FA ports to be used in a new masking view are chosen from the list provided in the EMC configuration file.

Storage group names

As volumes are attached to a host, they are either added to an existing storage group (if it exists) or a new storage group is created and the volume is then added. Storage groups contain volumes created from a pool (either single-pool or FAST-controlled), attached to a single host, over a single connection type (iSCSI or FC). Names are formed:

```
OS-[shortHostName][poolName]-I-SG (attached over iSCSI)
```

```
OS-[shortHostName][poolName]-F-SG (attached over Fibre Channel)
```

Concatenated or striped volumes

In order to support later expansion of created volumes, the VMAX Block Storage drivers create concatenated volumes as the default layout. If later expansion is not required, users can opt to create striped volumes in order to optimize I/O performance.

Below is an example of how to create striped volumes. First, create a volume type. Then define the extra spec for the volume type `storagetype:stripecount` representing the number of meta members in the striped volume. The example below means that each vol-

ume created under the GoldStriped volume type will be striped and made up of 4 meta members.

```
$ cinder type-create GoldStriped
$ cinder type-key GoldStriped set volume_backend_name=GOLD_BACKEND
$ cinder type-key GoldStriped set storagetype:stripecount=4
```

EMC VNX direct driver

EMC VNX direct driver (consists of EMCCLIISCSIDriver and EMCCLIFCDriver) supports both iSCSI and FC protocol. EMCCLIISCSIDriver (VNX iSCSI direct driver) and EMCCLIFCDriver (VNX FC direct driver) are separately based on the ISCSIDriver and FCDriver defined in Block Storage.

EMCCLIISCSIDriver and EMCCLIFCDriver perform the volume operations by executing Navisphere CLI (NaviSecCLI) which is a command line interface used for management, diagnostics and reporting functions for VNX.

Supported OpenStack release

EMC VNX direct driver supports the Kilo release.

System requirements

- VNX Operational Environment for Block version 5.32 or higher.
- VNX Snapshot and Thin Provisioning license should be activated for VNX.
- Navisphere CLI v7.32 or higher is installed along with the driver.

Supported 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.
- Clone a volume.
- Extend a volume.
- Migrate a volume.
- Retype a volume.
- Get volume statistics.
- Create and delete consistency groups.
- Create, list, and delete consistency group snapshots.
- Modify consistency groups.

Preparation

This section contains instructions to prepare the Block Storage nodes to use the EMC VNX direct driver. You install the Navisphere CLI, install the driver, ensure you have correct zoning configurations, and register the driver.

Install NaviSecCLI

Navisphere CLI needs to be installed on all Block Storage nodes within an OpenStack deployment.

- For Ubuntu x64, DEB is available at EMC OpenStack Github.
- For all other variants of Linux, Navisphere CLI is available at Downloads for VNX2 Series or Downloads for VNX1 Series.
- After installation, set the security level of Navisphere CLI to low:

```
$ /opt/Navisphere/bin/naviseccli security -certificate -setLevel low
```

Install Block Storage driver

Both `EMCCLIISCSIDriver` and `EMCCLIFCDriver` are provided in the installer package:

- `emc_vnx_cli.py`
- `emc_cli_fc.py` (for `EMCCLIFCDriver`)
- `emc_cli_iscsi.py` (for `EMCCLIISCSIDriver`)

Copy the files above to the `cinder/volume/drivers/emc/` directory of the OpenStack node(s) where `cinder-volume` is running.

FC zoning with VNX (`EMCCLIFCDriver` only)

A storage administrator must enable FC SAN auto zoning between all OpenStack nodes and VNX if FC SAN auto zoning is not enabled.

Register with VNX

Register the compute nodes with VNX to access the storage in VNX or enable initiator auto registration.

To perform "Copy Image to Volume" and "Copy Volume to Image" operations, the nodes running the `cinder-volume` service(Block Storage nodes) must be registered with the VNX as well.

Steps mentioned below are for a compute node. Please follow the same steps for the Block Storage nodes also. The steps can be skipped if initiator auto registration is enabled.

Note

When the driver notices that there is no existing storage group that has the host name as the storage group name, it will create the storage group and then add the compute nodes' or Block Storage nodes' registered initiators into the storage group.

If the driver notices that the storage group already exists, it will assume that the registered initiators have also been put into it and skip the operations above for better performance.

It is recommended that the storage administrator does not create the storage group manually and instead relies on the driver for the preparation. If the storage administrator needs to create the storage group manually for some special requirements, the correct registered initiators should be put into the storage group as well (otherwise the following volume attaching operations will fail).

EMCCLIFCDriver

Steps for `EMCCLIFCDriver`:

1. Assume `20:00:00:24:FF:48:BA:C2:21:00:00:24:FF:48:BA:C2` is the WWN of a FC initiator port name of the compute node whose hostname and IP are `myhost1` and `10.10.61.1`. Register `20:00:00:24:FF:48:BA:C2:21:00:00:24:FF:48:BA:C2` in Unisphere:

   a. Login to Unisphere, go to **FNM0000000000->Hosts->Initiators**.

   b. Refresh and wait until the initiator `20:00:00:24:FF:48:BA:C2:21:00:00:24:FF:48:BA:C2` with SP Port `A-1` appears.

   c. Click the **Register** button, select **CLARiiON/VNX** and enter the hostname (which is the output of the linux command `hostname`) and IP address:

      - Hostname : `myhost1`
      - IP : `10.10.61.1`
      - Click **Register**

   d. Then host `10.10.61.1` will appear under **Hosts->Host List** as well.

2. Register the wwn with more ports if needed.

EMCCLIISCSIDriver

Steps for `EMCCLIISCSIDriver`:

1. On the compute node with IP address `10.10.61.1` and hostname `myhost1`, execute the following commands (assuming `10.10.61.35` is the iSCSI target):

   a. Start the iSCSI initiator service on the node

   ```
   # /etc/init.d/open-iscsi start
   ```

   b. Discover the iSCSI target portals on VNX

   ```
   # iscsiadm -m discovery -t st -p 10.10.61.35
   ```

   c. Enter `/etc/iscsi`

```
# cd /etc/iscsi
```

d. Find out the iqn of the node

```
# more initiatorname.iscsi
```

2. Login to VNX from the compute 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
```

3. Assume `iqn.1993-08.org.debian:01:1a2b3c4d5f6g` is the initiator name of the compute node. Register `iqn.1993-08.org.debian:01:1a2b3c4d5f6g` in Unisphere:

   a. Login to Unisphere, go to **FNM0000000000->Hosts->Initiators** .

   b. Refresh and wait until the initiator `iqn.1993-08.org.debian:01:1a2b3c4d5f6g` with SP Port `A-8v0` appears.

   c. Click the **Register** button, select **CLARiiON/VNX** and enter the hostname (which is the output of the linux command `hostname`) and IP address:

   - Hostname : `myhost1`
   - IP : `10.10.61.1`
   - Click **Register**

   d. Then host `10.10.61.1` will appear under **Hosts->Host List** as well.

4. Logout iSCSI on the node:

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

5. Login to VNX from the compute node using the target corresponding to the SPB port:

```
# iscsiadm -m node -T iqn.1992-04.com.emc:cx.apm01234567890.b8 -p 10.10.
61.36 -l
```

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

7. Logout iSCSI on the node:

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

8. Register the iqn with more ports if needed.

Backend configuration

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

```
storage_vnx_pool_name = Pool_01_SAS
san_ip = 10.10.72.41
san_secondary_ip = 10.10.72.42
#VNX user name
#san_login = username
```

```
#VNX user password
#san_password = password
#VNX user type. Valid values are: global(default), local and ldap.
#storage_vnx_authentication_type = ldap
#Directory path of the VNX security file. Make sure the security file is
 generated first.
#VNX credentials are not necessary when using security file.
storage_vnx_security_file_dir = /etc/secfile/array1
naviseccli_path = /opt/Navisphere/bin/naviseccli
#timeout in minutes
default_timeout = 10
#If deploying EMCCLIISCSIDriver:
#volume_driver = cinder.volume.drivers.emc.emc_cli_iscsi.EMCCLIISCSIDriver
volume_driver = cinder.volume.drivers.emc.emc_cli_fc.EMCCLIFCDriver
destroy_empty_storage_group = False
#"node1hostname" and "node2hostname" shoule be the full hostnames of the
 nodes(Try command 'hostname').
#This option is for EMCCLIISCSIDriver only.
iscsi_initiators = {"node1hostname":["10.0.0.1", "10.0.0.2"],"node2hostname":
["10.0.0.3"]}

[database]
max_pool_size = 20
max_overflow = 30
```

- where `san_ip` is one of the SP IP addresses of the VNX array and `san_secondary_ip` is the other SP IP address of VNX array. `san_secondary_ip` is an optional field, and it serves the purpose of providing a high availability(HA) design. In case that one SP is down, the other SP can be connected automatically. `san_ip` is a mandatory field, which provides the main connection.

- where `Pool_01_SAS` is the pool from which the user wants to create volumes. The pools can be created using Unisphere for VNX. Refer to the the section called "Multiple pools support" [40] on how to manage multiple pools.

- where `storage_vnx_security_file_dir` is the directory path of the VNX security file. Make sure the security file is generated following the steps in the section called "Authentication" [35].

- where `iscsi_initiators` is a dictionary of IP addresses of the iSCSI initiator ports on all OpenStack nodes which want to connect to VNX via iSCSI. If this option is configured, the driver will leverage this information to find an accessible iSCSI target portal for the initiator when attaching volume. Otherwise, the iSCSI target portal will be chosen in a relative random way.

- Restart `cinder-volume` service to make the configuration change take effect.

Authentication

VNX credentials are necessary when the driver connects to the VNX system. Credentials in global, local and ldap scopes are supported. There are two approaches to provide the credentials.

The recommended one is using the Navisphere CLI security file to provide the credentials which can get rid of providing the plain text credentials in the configuration file. Following is the instruction on how to do this.

1. Find out the linux user id of the `/usr/bin/cinder-volume` processes. Assuming the service `/usr/bin/cinder-volume` is running by account `cinder`.

2. Switch to `root` account

3. Change `cinder:x:113:120::/var/lib/cinder:/bin/false` to `cinder:x:113:120::/var/lib/cinder:/bin/bash` in `/etc/passwd` (This temporary change is to make step 4 work).

4. Save the credentials on behalf of `cinder` user to a security file (assuming the array credentials are `admin/admin` in `global` scope). In below command, switch `-secfilepath` is used to specify the location to save the security file (assuming saving to directory `/etc/secfile/array1`).

   ```
   # su -l cinder -c '/opt/Navisphere/bin/naviseccli -AddUserSecurity -user
    admin -password admin -scope 0 -secfilepath /etc/secfile/array1'
   ```

   Save the security file to the different locations for different arrays except where the same credentials are shared between all arrays managed by the host. Otherwise, the credentials in the security file will be overwritten. If `-secfilepath` is not specified in the command above, the security file will be saved to the default location which is the home directory of the executor.

5. Change `cinder:x:113:120::/var/lib/cinder:/bin/bash` back to `cinder:x:113:120::/var/lib/cinder:/bin/false` in `/etc/passwd`.

6. Remove the credentials options `san_login`, `san_password` and `storage_vnx_authentication_type` from `cinder.conf` (normally it is `/etc/cinder/cinder.conf`). Add the option `storage_vnx_security_file_dir` and set its value to the directory path supplied with switch `-secfilepath` in step 4. Omit this option if `-secfilepath` is not used in step 4.

   ```
   #Directory path that contains the VNX security file. Generate the security
    file first
   storage_vnx_security_file_dir = /etc/secfile/array1
   ```

7. Restart `cinder-volume` service to make the change take effect.

Alternatively, the credentials can be specified in `/etc/cinder/cinder.conf` through the three options below:

```
#VNX user name
san_login = username
#VNX user password
san_password = password
#VNX user type. Valid values are: global, local and ldap. global is the
 default value
storage_vnx_authentication_type = ldap
```

Restriction of deployment

It does not suggest to deploy the driver on a compute node if `cinder upload-to-image --force True` is used against an in-use volume. Otherwise, `cinder upload-to-image --force True` will terminate the vm instance's data access to the volume.

Restriction of volume extension

VNX does not support to extend the thick volume which has a snapshot. If the user tries to extend a volume which has a snapshot, the volume's status would change to `error_extending`.

Restriction of iSCSI attachment

The driver caches the iSCSI ports information. If the iSCSI port configurations are changed, the administrator should restart the `cinder-volume` service or wait 5 minutes before any volume attachment operation. Otherwise, the attachment may fail because the old iSCSI port configurations were used.

Provisioning type (thin, thick, deduplicated and compressed)

User can specify extra spec key `storagetype:provisioning` in volume type to set the provisioning type of a volume. The provisioning type can be `thick`, `thin`, `deduplicated` or `compressed`.

- `thick` provisioning type means the volume is fully provisioned.
- `thin` provisioning type means the volume is virtually provisioned.
- `deduplicated` provisioning type means the volume is virtually provisioned and the deduplication is enabled on it. Administrator shall go to VNX to configure the system level deduplication settings. To create a deduplicated volume, the VNX deduplication license should be activated on VNX first, and use key `deduplication_support=True` to let Block Storage scheduler find a volume back end which manages a VNX with deduplication license activated.
- `compressed` provisioning type means the volume is virtually provisioned and the compression is enabled on it. Administrator shall go to the VNX to configure the system level compression settings. To create a compressed volume, the VNX compression license should be activated on VNX first, and the user should specify key `compression_support=True` to let Block Storage scheduler find a volume back end which manages a VNX with compression license activated. VNX does not support to create a snapshot on a compressed volume. If the user tries to create a snapshot on a compressed volume, the operation would fail and OpenStack would show the new snapshot in error state.

Here is an example about how to create a volume with provisioning type. Firstly create a volume type and specify storage pool in the extra spec, then create a volume with this volume type:

```
$ cinder type-create "ThickVolume"
$ cinder type-create "ThinVolume"
$ cinder type-create "DeduplicatedVolume"
$ cinder type-create "CompressedVolume"
$ cinder type-key "ThickVolume" set storagetype:provisioning=thick
$ cinder type-key "ThinVolume" set storagetype:provisioning=thin
$ cinder type-key "DeduplicatedVolume" set storagetype:provisioning=
deduplicated deduplication_support=True
```

```
$ cinder type-key "CompressedVolume" set storagetype:provisioning=compressed
 compression_support=True
```

In the example above, four volume types are created: `ThickVolume`, `ThinVolume`, `DeduplicatedVolume` and `CompressedVolume`. For `ThickVolume`, `storagetype:provisioning` is set to `thick`. Similarly for other volume types. If `storagetype:provisioning` is not specified or an invalid value, the default value `thick` is adopted.

Volume type name, such as `ThickVolume`, is user-defined and can be any name. Extra spec key `storagetype:provisioning` shall be the exact name listed here. Extra spec value for `storagetype:provisioning` shall be `thick`, `thin`, `deduplicated` or `compressed`. During volume creation, if the driver finds `storagetype:provisioning` in the extra spec of the volume type, it will create the volume with the provisioning type accordingly. Otherwise, the volume will be thick as the default.

Fully automated storage tiering support

VNX supports Fully automated storage tiering which requires the FAST license activated on the VNX. The OpenStack administrator can use the extra spec key `storagetype:tiering` to set the tiering policy of a volume and use the extra spec key `fast_support=True` to let Block Storage scheduler find a volume back end which manages a VNX with FAST license activated. Here are the five supported values for the extra spec key `storagetype:tiering`:

- `StartHighThenAuto` (Default option)
- `Auto`
- `HighestAvailable`
- `LowestAvailable`
- `NoMovement`

Tiering policy can not be set for a deduplicated volume. The user can check storage pool properties on VNX to know the tiering policy of a deduplicated volume.

Here is an example about how to create a volume with tiering policy:

```
$ cinder type-create "AutoTieringVolume"
$ cinder type-key "AutoTieringVolume" set storagetype:tiering=Auto
 fast_support=True
$ cinder type-create "ThinVolumeOnLowestAvaibleTier"
$ cinder type-key "CompressedVolumeOnLowestAvaibleTier" set
 storagetype:provisioning=thin storagetype:tiering=Auto fast_support=True
```

FAST Cache support

VNX has FAST Cache feature which requires the FAST Cache license activated on the VNX. The OpenStack administrator can use the extra spec key `fast_cache_enabled` to choose whether to create a volume on the volume back end which manages a pool with FAST Cache enabled. The value of the extra spec key `fast_cache_enabled` is either `True` or `False`. When creating a volume, if the key `fast_cache_enabled` is set to `True` in the

volume type, the volume will be created by a back end which manages a pool with FAST Cache enabled.

Storage group automatic deletion

For volume attaching, the driver has a storage group on VNX for each compute node hosting the vm instances that are going to consume VNX Block Storage (using the compute node's hostname as the storage group's name). All the volumes attached to the vm instances in a computer node will be put into the corresponding Storage Group. If `destroy_empty_storage_group=True`, the driver will remove the empty storage group when its last volume is detached. For data safety, it does not suggest to set the option `destroy_empty_storage_group=True` unless the VNX is exclusively managed by one Block Storage node because consistent `lock_path` is required for operation synchronization for this behavior.

EMC storage-assisted volume migration

`EMC VNX direct driver` supports storage-assisted volume migration, when the user starts migrating with `cinder migrate --force-host-copy False volume_id host` or `cinder migrate volume_id host`, cinder will try to leverage the VNX's native volume migration functionality.

In the following scenarios, VNX native volume migration will not be triggered:

- Volume migration between back ends with different storage protocol, ex, FC and iSCSI.
- Volume is being migrated across arrays.

Initiator auto registration

If `initiator_auto_registration=True`, the driver will automatically register iSCSI initiators with all working iSCSI target ports on the VNX array during volume attaching (The driver will skip those initiators that have already been registered).

If the user wants to register the initiators with some specific ports on VNX but not register with the other ports, this functionality should be disabled.

Initiator auto deregistration

Enabling storage group automatic deletion is the precondition of this functionality. If `initiator_auto_deregistration=True` is set, the driver will deregister all the iSCSI initiators of the host after its storage group is deleted.

Read-only volumes

OpenStack supports read-only volumes. The following command can be used to set a volume to read-only.

```
$ cinder readonly-mode-update volume True
```

After a volume is marked as read-only, the driver will forward the information when a hypervisor is attaching the volume and the hypervisor will have an implementation-specific way to make sure the volume is not written.

Multiple pools support

Normally the user configures a storage pool for a Block Storage back end (named as pool-based back end), so that the Block Storage back end uses only that storage pool.

If `storage_vnx_pool_name` is not given in the configuration file, the Block Storage back end uses all the pools on the VNX array, and the scheduler chooses the pool to place the volume based on its capacities and capabilities. This kind of Block Storage back end is named as array-based back end.

Here is an example about configuration of array-based back end:

```
san_ip = 10.10.72.41
#Directory path that contains the VNX security file. Make sure the security
 file is generated first
storage_vnx_security_file_dir = /etc/secfile/array1
storage_vnx_authentication_type = global
naviseccli_path = /opt/Navisphere/bin/naviseccli
default_timeout = 10
volume_driver = cinder.volume.drivers.emc.emc_cli_iscsi.EMCCLIISCSIDriver
destroy_empty_storage_group = False
volume_backend_name = vnx_41
```

In this configuration, if the user wants to create a volume on a certain storage pool, a volume type with a extra spec specified the storage pool should be created first, then the user can use this volume type to create the volume.

Here is an example about creating the volume type:

```
$ cinder type-create "HighPerf"
$ cinder type-key "HighPerf" set pool_name=Pool_02_SASFLASH
 volume_backend_name=vnx_41
```

Volume number threshold

In VNX, there is a limit on the maximum number of pool volumes that can be created in the system. When the limit is reached, no more pool volumes can be created even if there is enough remaining capacity in the storage pool. In other words, if the scheduler dispatches a volume creation request to a back end that has free capacity but reaches the limit, the back end will fail to create the corresponding volume.

The default value of the option `check_max_pool_luns_threshold` is `False`. When `check_max_pool_luns_threshold=True`, the pool-based back end will check the limit and will report 0 free capacity to the scheduler if the limit is reached. So the scheduler will be able to skip this kind of pool-based back end that runs out of the pool volume number.

FC SAN auto zoning

EMC direct driver supports FC SAN auto zoning when ZoneManager is configured. Set `zoning_mode` to `fabric` in back-end configuration section to enable this feature. For ZoneManager configuration, please refer to the section called “Fibre Channel Zone Manager” [177].

Multi-backend configuration

```
[DEFAULT]
```

```
enabled_backends = backendA, backendB

[backendA]

storage_vnx_pool_name = Pool_01_SAS
san_ip = 10.10.72.41
#Directory path that contains the VNX security file. Make sure the security
 file is generated first.
storage_vnx_security_file_dir = /etc/secfile/array1
naviseccli_path = /opt/Navisphere/bin/naviseccli
#Timeout in Minutes
default_timeout = 10
volume_driver = cinder.volume.drivers.emc.emc_cli_fc.EMCCLIFCDriver
destroy_empty_storage_group = False
initiator_auto_registration = True

[backendB]
storage_vnx_pool_name = Pool_02_SAS
san_ip = 10.10.26.101
san_login = username
san_password = password
naviseccli_path = /opt/Navisphere/bin/naviseccli
#Timeout in Minutes
default_timeout = 10
volume_driver = cinder.volume.drivers.emc.emc_cli_fc.EMCCLIFCDriver
destroy_empty_storage_group = False
initiator_auto_registration = True

[database]

max_pool_size = 20
max_overflow = 30
```

For more details on multi-backend, see OpenStack Cloud Administration Guide.

Force delete volumes in storage groups

Some available volumes may remain in storage groups on the VNX array due to some OpenStack timeout issues. But the VNX array does not allow the user to delete the volumes which are still in storage groups. The option `force_delete_lun_in_storagegroup` is introduced to allow the user to delete the available volumes in this tricky situation.

When `force_delete_lun_in_storagegroup=True` is set in the back-end section, the driver will move the volumes out of storage groups and then delete them if the user tries to delete the volumes that remain in storage groups on the VNX array.

The default value of `force_delete_lun_in_storagegroup` is `False`.

EMC XtremIO Block Storage driver configuration

The high performance XtremIO All Flash Array (AFA) offers Block Storage services to OpenStack. Using the driver, OpenStack Block Storage hosts can connect to an XtermIO Storage cluster.

This section explains how to configure and connect an OpenStack block storage host to an XtremIO storage cluster.

Support matrix

- Xtremapp: Version 3.0 and 4.0

Supported operations

- Create, delete, clone, attach, and detach volumes
- Create and delete volume snapshots
- Create a volume from a snapshot
- Copy an image to a volume
- Copy a volume to an image
- Extend a volume
- Manage and unmanage a volume
- Get volume statistics

XtremIO Block Storage driver configuration

Edit the `cinder.conf` file by adding the configuration below under the `[DEFAULT]` section of the file in case of a single back end or under a separate section in case of multiple back ends (for example [XTREMIO]). The configuration file is usually located under the following path `/etc/cinder/cinder.conf`.

For a configuration example, refer to the configuration example.

XtremIO driver name

Configure the driver name by adding the following parameter:

- For iSCSI `volume_driver = cinder.volume.drivers.emc.xtremio.XtremIOIscsiDriver`
- For Fibre Channel `volume_driver = cinder.volume.drivers.emc.xtremio.XtremIOFibreChannelDriver`

XtremIO management server (XMS) IP

To retrieve the management IP, use the **show-xms** CLI command.

Configure the management IP by adding the following parameter: `san_ip = XMS Management IP`

XtremIO cluster name

In XtremIO version 4.0, a single XMS can manage multiple cluster back ends. In such setups, the administrator is required to specify the cluster name (in addition to the XMS IP). Each cluster must be defined as a separate back end.

To retrieve the Cluster Name, run the **show-clusters** CLI command.

Configure the cluster name by adding the `xtremio_cluster_name = Cluster-Name`

Note

When a single cluster is managed in XtremIO version 4.0, the cluster name is not required.

XtremIO user credentials

OpenStack Block Storage requires an XtremIO XMS user with administrative privileges. XtremIO recommends creating a dedicated OpenStack user account that holds an administrative user role.

Refer to the *XtremIO User Guide* for details on user account management

Create an XMS account using either the XMS GUI or the **add-user-account**CLI command.

Configure the user credentials by adding the following parameters:

`san_login = XMS username`

`san_password = XMS username password`

Multiple back ends

Configuring multiple storage back ends enables you to create several back-end storage solutions that serve the same OpenStack Compute resources.

When a volume is created, the scheduler selects the appropriate back end to handle the request, according to the specified volume type.

Setting thin provisioning and multipathing parameters

To support thin provisioning and multipathing in the XtremIO Array, the following parameters from the Nova and Cinder configuration files should be modified as follows:

- Thin Provisioning

  All XtremIO volumes are thin provisioned. The default value of 20 should be maintained for the `max_over_subscription_ratio` parameter.

  The `use_cow_images` parameter in the`nova.conf`file should be set to False as follows:

  `use_cow_images = false`

- Multipathing

  The `use_multipath_for_image_xfer` parameter in the`cinder.conf` file should be set to True as follows:

  `use_multipath_for_image_xfer = true`

Restarting OpenStack Block Storage

Save the `cinder.conf` file and restart cinder by running the following command:

```
$ openstack-service restart cinder-volume
```

Configuring CHAP

The XtremIO Block Storage driver supports CHAP initiator authentication. If CHAP initiator authentication is required, set the CHAP Authentication mode to initiator.

To set the CHAP initiator mode using CLI, run the following CLI command:

```
$ modify-chap chap-authentication-mode=initiator
```

The CHAP initiator mode can also be set via the XMS GUI

Refer to *XtremIO User Guide* for details on CHAP configuration via GUI and CLI.

The CHAP initiator authentication credentials (username and password) are generated automatically by the Block Storage driver. Therefore, there is no need to configure the initial CHAP credentials manually in XMS.

Configuration example

cinder.conf example file

You can update the `cinder.conf` file by editing the necessary parameters as follows:

```
[Default]
enabled_backends = XtremIO

[XtremIO]
volume_driver = cinder.volume.drivers.emc.xtremio.XtremIOFibreChannelDriver
san_ip = XMS_IP
xtremio_cluster_name = Cluster01
san_login = XMS_USER
san_password = XMS_PASSWD
volume_backend_name = XtremIOAFA
```

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 the use of GlusterFS in a similar fashion as NFS. It supports basic volume operations, including 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 2.4. Description of GlusterFS storage configuration options

Configuration option = Default value	Description
[DEFAULT]	
`glusterfs_mount_point_base` = *`$state_path/mnt`*	(StrOpt) Base dir containing mount points for gluster shares.
`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 HNAS iSCSI and NFS driver

This OpenStack Block Storage volume driver provides iSCSI and NFS support for Hitachi NAS Platform Models 3080, 3090, 4040, 4060, 4080 and 4100.

Supported operations

The NFS and iSCSI drivers support 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.
- Extend a volume.
- Get volume statistics.

HNAS storage requirements

Before using iSCSI and NFS services, use the HNAS configuration and management GUI (SMU) or SSC CLI to create storage pool(s), file system(s), and assign an EVS. Make sure that the file system used is not created as `replication targets`. Additionally:

For NFS: Create NFS exports, choose a path for them (it must be different from "/") and set the **Show snapshots** option to `hide and disable access`.

Also, configure the option `norootsquash` as `"* (rw, noroot-squash)"`, so cinder services can change the permissions of its volumes.

In order to use the hardware accelerated features of NFS HNAS, we recommend setting `max-nfs-version` to 3. Refer to HNAS command line reference to see how to configure this option.

For iSCSI: You need to set an iSCSI domain.

Block storage host requirements

The HNAS driver is supported for Red Hat, SUSE Cloud and Ubuntu Cloud. The following packages must be installed:

1. nfs-utils for Red Hat
2. nfs-client for SUSE
3. nfs-common, libc6-i386 for Ubuntu (libc6-i386 only required on Ubuntu 12.04)
4. If you are not using SSH, you need the HDS SSC package (hds-ssc-v1.0-1) to communicate with an HNAS array using the **SSC** command. This utility package is available in the RPM package distributed with the hardware through physical media or it can be manually copied from the SMU to the Block Storage host.

Package installation

If you are installing the driver from a RPM or DEB package, follow the steps bellow:

1. Install SSC:

   In Red Hat:

   ```
   # rpm -i hds-ssc-v1.0-1.rpm
   ```

   Or in SUSE:

   ```
   # zypper hds-ssc-v1.0-1.rpm
   ```

   Or in Ubuntu:

   ```
   # dpkg -i hds-ssc_1.0-1_all.deb
   ```

2. Install the dependencies:

   In Red Hat:

   ```
   # yum install nfs-utils nfs-utils-lib
   ```

   Or in Ubuntu:

   ```
   # apt-get install nfs-common
   ```

   Or in SUSE:

   ```
   # zypper install nfs-client
   ```

If you are using Ubuntu 12.04, you also need to install libc6-i386

```
# apt-get install libc6-i386
```

3. Configure the driver as described in the "Driver Configuration" section.

4. Restart all cinder services (volume, scheduler and backup).

Driver configuration

The HDS driver supports the concept of differentiated services (also referred as quality of service) by mapping volume types to services provided through HNAS.

HNAS supports a variety of storage options and file system capabilities, which are selected through the definition of volume types and the use of multiple back ends. The driver maps up to four volume types into separated exports or file systems, and can support any number if using multiple back ends.

The configuration for the driver is read from an XML-formatted file (one per back end), which you need to create and set its path in the `cinder.conf` configuration file. Below are the configuration needed in the `cinder.conf` configuration file [1]:

```
[DEFAULT]
enabled_backends = hnas_iscsi1, hnas_nfs1
```

For HNAS iSCSI driver create this section:

```
[hnas_iscsi1]
volume_driver = cinder.volume.drivers.hds.iscsi.HDSISCSIDriver
hds_hnas_iscsi_config_file = /path/to/config/hnas_config_file.xml
volume_backend_name = HNAS-ISCSI
```

For HNAS NFS driver create this section:

```
[hnas_nfs1]
volume_driver = cinder.volume.drivers.hds.nfs.HDSNFSDriver
hds_hnas_nfs_config_file = /path/to/config/hnas_config_file.xml
volume_backend_name = HNAS-NFS
```

The XML file has the following format:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
  <config>
    <mgmt_ip0>172.24.44.15</mgmt_ip0>
    <hnas_cmd>ssc</hnas_cmd>
    <chap_enabled>False</chap_enabled>
    <ssh_enabled>False</ssh_enabled>
    <cluster_admin_ip0>10.1.1.1</cluster_admin_ip0>
    <username>supervisor</username>
    <password>supervisor</password>
    <svc_0>
      <volume_type>default</volume_type>
      <iscsi_ip>172.24.44.20</iscsi_ip>
      <hdp>fs01-husvm</hdp>
```

[1] The configuration file location may differ.

```
    </svc_0>
    <svc_1>
      <volume_type>platinun</volume_type>
      <iscsi_ip>172.24.44.20</iscsi_ip>
      <hdp>fs01-platinun</hdp>
    </svc_1>
  </config>
```

HNAS volume driver XML configuration options

An OpenStack Block Storage node using HNAS drivers can have up to four services. Each service is defined by a `svc_n` tag (`svc_0`, `svc_1`, `svc_2`, or `svc_3` [2], for example). These are the configuration options available for each service label:

Table 2.5. Configuration options for service labels

Option	Type	Default	Description
`volume_type`	Required	`default`	When a `create_volume` call with a certain volume type happens, the volume type will try to be matched up with this tag. In each configuration file you must define the `default` volume type in the service labels and, if no volume type is specified, the `default` is used. Other labels are case sensitive and should match exactly. If no configured volume types match the incoming requested type, an error occurs in the volume creation.
`iscsi_ip`	Required only for iSCSI		An iSCSI IP address dedicated to the service.
`hdp`	Required		For iSCSI driver: virtual file system label associated with the service. For NFS driver: path to the volume (<ip_address>:/<path>) associated with the service. Additionally, this entry must be added in the file used to list available NFS shares. This file is located, by default, in `/etc/cinder/nfs_shares` or you can specify the location in the `nfs_shares_config` option in the `cinder.conf` configuration file.

These are the configuration options available to the `config` section of the XML config file:

Table 2.6. Configuration options

Option	Type	Default	Description
`mgmt_ip0`	Required		Management Port 0 IP address. Should be the IP address of the "Admin" EVS.
`hnas_cmd`	Optional	ssc	Command to communicate to HNAS array.
`chap_enabled`	Optional (iSCSI only)	`True`	Boolean tag used to enable CHAP authentication protocol.
`username`	Required	supervisor	It's always required on HNAS.
`password`	Required	supervisor	Password is always required on HNAS.
`svc_0, svc_1, svc_2, svc_3`	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 HDP and a unique volume type.

[2] There is no relative precedence or weight among these four labels.

Option	Type	Default	Description
cluster_admin_ip0	Optional if ssh_enabled is True		The address of HNAS cluster admin.
ssh_enabled	Optional	False	Enables SSH authentication between Block Storage host and the SMU.
ssh_private_key	Required if ssh_enabled is True	False	Path to the SSH private key used to authenticate in HNAS SMU. The public key must be uploaded to HNAS SMU using ssh-register-public-key (this is an SSH sub-command). Note that copying the public key HNAS using ssh-copy-id doesn't work properly as the SMU periodically wipe out those keys.

Service labels

HNAS driver supports differentiated types of service using the service labels. It is possible to create up to four types of them, as gold, platinun, silver and ssd, for example.

After creating the services in the XML configuration file, you must configure one volume_type per service. Each volume_type must have the metadata service_label with the same name configured in the <volume_type> section of that service. If this is not set, OpenStack Block Storage will schedule the volume creation to the pool with largest available free space or other criteria configured in volume filters.

```
$ cinder type-create 'default'
$ cinder type-key 'default' set service_label = 'default'
$ cinder type-create 'platinun-tier'
$ cinder type-key 'platinun' set service_label = 'platinun'
```

Multi-back-end configuration

If you use multiple back ends and intend to enable the creation of a volume in a specific back end, you must configure volume types to set the volume_backend_name option to the appropriate back end. Then, create volume_type configurations with the same volume_backend_name .

```
$ cinder type-create 'iscsi'
$ cinder type-key 'iscsi' set volume_backend_name = 'HNAS-ISCSI'
$ cinder type-create 'nfs'
$ cinder type-key 'nfs' set volume_backend_name = 'HNAS-NFS'
```

You can deploy multiple OpenStack HNAS drivers instances that each control a separate HNAS array. Each service (svc_0, svc_1, svc_2, svc_3) on the instances need to have a volume_type and service_label metadata associated with it. If no metadata is associated with a pool, OpenStack Block Storage filtering algorithm selects the pool with the largest available free space.

SSH configuration

Instead of using **SSC** on the Block Storage host and store its credential on the XML configuration file, HNAS driver supports **SSH** authentication. To configure that:

1. If you don't have a pair of public keys already generated, create it in the Block Storage host (leave the pass-phrase empty):

   ```
   $ mkdir -p /opt/hds/ssh
   ```

```
$ ssh-keygen -f /opt/hds/ssh/hnaskey
```

2. Change the owner of the key to `cinder` (or the user the volume service will be run):

```
# chown -R cinder.cinder /opt/hds/ssh
```

3. Create the directory "ssh_keys" in the SMU server:

```
$ ssh [manager|supervisor]@<smu-ip> 'mkdir -p /var/opt/mercury-main/home/
[manager|supervisor]/ssh_keys/'
```

4. Copy the public key to the "ssh_keys" directory:

```
$ scp /opt/hds/ssh/hnaskey.pub [manager|supervisor]@<smu-ip>:/var/opt/
mercury-main/home/[manager|supervisor]/ssh_keys/
```

5. Access the SMU server:

```
$ ssh [manager|supervisor]@<smu-ip>
```

6. Run the command to register the SSH keys:

```
$ ssh-register-public-key -u [manager|supervisor] -f ssh_keys/hnaskey.pub
```

7. Check the communication with HNAS in the Block Storage host:

```
$ ssh -i /opt/hds/ssh/hnaskey [manager|supervisor]@<smu-ip> 'ssc
 <cluster_admin_ip0> df -a'
```

`<cluster_admin_ip0>` is "localhost" for single node deployments. This should return a list of available file systems on HNAS.

Editing the XML config file:

1. Set the "username".
2. Enable SSH adding the line "`<ssh_enabled> True</ssh_enabled>`" under "`<config>`" session.
3. Set the private key path: "`<ssh_private_key> /opt/hds/ssh/hnaskey</ssh_private_key>`" under "`<config>`" session.
4. If the HNAS is in a multi-cluster configuration set "`<cluster_admin_ip0>`" to the cluster node admin IP. In a single node HNAS, leave it empty.
5. Restart the cinder service.

Additional notes

- The `get_volume_stats()` function always provides the available capacity based on the combined sum of all the HDPs that are used in these services labels.
- After changing the configuration on the storage, the OpenStack Block Storage driver must be restarted.
- HNAS iSCSI driver, due to an HNAS limitation, allows only 32 volumes per target.

- On Red Hat, if the system is configured to use SELinux, you need to set "`virt_use_nfs = on`" for NFS driver work properly.

Hitachi storage volume driver

Hitachi storage volume driver provides iSCSI and Fibre Channel support for Hitachi storages.

System requirements

Supported storages:

- Hitachi Virtual Storage Platform G1000 (VSP G1000)
- Hitachi Virtual Storage Platform (VSP)
- Hitachi Unified Storage VM (HUS VM)
- Hitachi Unified Storage 100 Family (HUS 100 Family)

Required software:

- RAID Manager Ver 01-32-03/01 or later for VSP G1000/VSP/HUS VM
- Hitachi Storage Navigator Modular 2 (HSNM2) Ver 27.50 or later for HUS 100 Family

Note

HSNM2 needs to be installed under `/usr/stonavm`.

Required licenses:

- Hitachi In-System Replication Software for VSP G1000/VSP/HUS VM
- (Mandatory) ShadowImage in-system replication for HUS 100 Family
- (Optional) Copy-on-Write Snapshot for HUS 100 Family

Additionaly, the pexpect package is required.

Supported operations

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

- Extend a volume.
- Get volume statistics.

Configuration

Set up Hitachi storage

You need to specify settings as described below. For details about each step, see the user's guide of the storage device. Use a storage administrative software such as Storage Navigator to set up the storage device so that LDEVs and host groups can be created and deleted, and LDEVs can be connected to the server and can be asynchronously copied.

1. Create a Dynamic Provisioning pool.
2. Connect the ports at the storage to the Controller node and Compute nodes.
3. For VSP G1000/VSP/HUS VM, set "port security" to "enable" for the ports at the storage.
4. For HUS 100 Family, set "Host Group security"/"iSCSI target security" to "ON" for the ports at the storage.
5. For the ports at the storage, create host groups (iSCSI targets) whose names begin with HBSD- for the Controller node and each Compute node. Then register a WWN (initiator IQN) for each of the Controller node and Compute nodes.
6. For VSP G1000/VSP/HUS VM, perform the following:
   - Create a storage device account belonging to the Administrator User Group. (To use multiple storage devices, create the same account name for all the target storage devices, and specify the same resource group and permissions.)
   - Create a command device (In-Band), and set user authentication to ON.
   - Register the created command device to the host group for the Controller node.
   - To use the Thin Image function, create a pool for Thin Image.
7. For HUS 100 Family, perform the following:
   - Use the command **auunitaddauto** to register the unit name and controller of the storage device to HSNM2.
   - When connecting via iSCSI, if you are using CHAP certification, specify the same user and password as that used for the storage port.

Set up Hitachi Gigabit Fibre Channel adaptor

Change a parameter of the hfcldd driver and update the initram file if Hitachi Gigabit Fibre Channel adaptor is used.

```
# /opt/hitachi/drivers/hba/hfcmgr -E hfc_rport_lu_scan 1
# dracut -f initramfs-KERNEL_VERSION.img KERNEL_VERSION
# reboot
```

Set up Hitachi storage volume driver

1. Create directory.

```
# mkdir /var/lock/hbsd
# chown cinder:cinder /var/lock/hbsd
```

2. Create "volume type" and "volume key".

   This example shows that HUS100_SAMPLE is created as "volume type" and hus100_backend is registered as "volume key".

```
$ cinder type-create HUS100_SAMPLE
$ cinder type-key HUS100_SAMPLE set volume_backend_name=hus100_backend
```

   Please specify any identical "volume type" name and "volume key".

   To confirm the created "volume type", please execute the following command:

```
$ cinder extra-specs-list
```

3. Edit `/etc/cinder/cinder.conf` as follows.

   If you use Fibre Channel:

```
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
```

   If you use iSCSI:

```
volume_driver = cinder.volume.drivers.hitachi.hbsd_iscsi.HBSDISCSIDriver
```

   Also, set `volume_backend_name` created by **cinder type-key**

```
volume_backend_name = hus100_backend
```

   This table shows configuration options for Hitachi storage volume driver.

Table 2.7. Description of Hitachi storage volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`hitachi_add_chap_user` = *False*	(BoolOpt) Add CHAP user
`hitachi_async_copy_check_interval` = *10*	(IntOpt) Interval to check copy asynchronously
`hitachi_auth_method` = *None*	(StrOpt) iSCSI authentication method
`hitachi_auth_password` = *HBSD-CHAP-password*	(StrOpt) iSCSI authentication password
`hitachi_auth_user` = *HBSD-CHAP-user*	(StrOpt) iSCSI authentication username
`hitachi_copy_check_interval` = *3*	(IntOpt) Interval to check copy
`hitachi_copy_speed` = *3*	(IntOpt) Copy speed of storage system
`hitachi_default_copy_method` = *FULL*	(StrOpt) Default copy method of storage system
`hitachi_group_range` = *None*	(StrOpt) Range of group number
`hitachi_group_request` = *False*	(BoolOpt) Request for creating HostGroup or iSCSI Target
`hitachi_horcm_add_conf` = *True*	(BoolOpt) Add to HORCM configuration

Configuration option = Default value	Description
hitachi_horcm_numbers = *200,201*	(StrOpt) Instance numbers for HORCM
hitachi_horcm_password = *None*	(StrOpt) Password of storage system for HORCM
hitachi_horcm_user = *None*	(StrOpt) Username of storage system for HORCM
hitachi_ldev_range = *None*	(StrOpt) Range of logical device of storage system
hitachi_pool_id = *None*	(IntOpt) Pool ID of storage system
hitachi_serial_number = *None*	(StrOpt) Serial number of storage system
hitachi_target_ports = *None*	(StrOpt) Control port names for HostGroup or iSCSI Target
hitachi_thin_pool_id = *None*	(IntOpt) Thin pool ID of storage system
hitachi_unit_name = *None*	(StrOpt) Name of an array unit
hitachi_zoning_request = *False*	(BoolOpt) Request for FC Zone creating HostGroup

4. Restart Block Storage service.

   When the startup is done, "MSGID0003-I: The storage backend can be used." is output into `/var/log/cinder/volume.log` as follows.

```
2014-09-01 10:34:14.169 28734 WARNING cinder.volume.drivers.hitachi.
hbsd_common [req-a0bb70b5-7c3f-422a-a29e-6a55d6508135 None None]
MSGID0003-I: The storage backend can be used. (config_group:
hus100_backend)
```

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:

- HP 3PAR Operating System software version 3.1.3 MU1 or higher.
  - Deduplication provisioning requires SSD disks and HP 3PAR Operating System software version 3.2.1 MU1 or higher.
  - Enabling Flash Cache Policy requires the following:
    - Array must contain SSD disks.
    - HP 3PAR Operating System software version 3.2.1 MU2 or higher.
    - hp3parclient version 3.2.0 or newer.
    - Array must have the Adaptive Flash Cache license installed.

- Flash Cache must be enabled on the array with the CLI command **createflashcache size**, where size must be in 16 GB increments. For example, **createflashcache 128g** will create 128 GB of Flash Cache for each node pair in the array.

- The Dynamic Optimization license is required to support any feature that results in a volume changing provisioning type or CPG. This may apply to the volume **migrate**, **retype**, and **manage** commands.

- HP 3PAR Web Services API Server must be enabled and running
- One Common Provisioning Group (CPG)
- Additionally, you must install the hp3parclient version 3.1.1 or newer from the Python standard library on the system with the enabled Block Storage service volume drivers.

Supported 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.
- Extend a volume.
- Migrate a volume with back-end assistance.
- Retype a volume.
- Manage and unmanage a volume.

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`

Warning

This setting is ignored as of Kilo. Instead, use the `hp3par_cpg` setting in `cinder.conf` to list the valid CPGs for a backend. CPGs should now be controlled by configuring separate backends with pools.

- `hp3par:snap_cpg`
- `hp3par:provisioning`
- `hp3par:persona`

- `hp3par:vvs`
- `hp3par:flash_cache`

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`, `full`, and `dedup`.
- `hp3par:persona` - Defaults to the `2 - Generic-ALUA` persona. The valid values are, `1 - Generic`, `2 - Generic-ALUA`, `3 - Generic-legacy`, `4 - HPUX-legacy`, `5 - AIX-legacy`, `6 - EGENERA`, `7 - ONTAP-legacy`, `8 - VMware`, `9 - OpenVMS`, `10 - HPUX`, and `11 - WindowsServer`.
- `hp3par:flash_cache` - Defaults to `false`, the valid values are `true` and `false`.

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.

The following keys require that the HP 3PAR StoreServ storage array has an Adaptive Flash Cache license installed.

- `hp3par:flash_cache` - The flash-cache policy, which can be turned on and off by setting the value to `true` or `false`.

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.

```
# showwsapi
-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 username with the 'edit' role
hp3par_username=edit3par

# 3PAR password for the user specified in hp3par_username
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

# Enable CHAP authentication for iSCSI connections.
hp3par_iscsi_chap_enabled=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) plug-in 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 ar-

ray, 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" [60]. To configure the driver in legacy mode, see the section called "HP LeftHand/StoreVirtual CLIQ driver legacy mode" [63].

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, 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.
- Extend a volume.
- Get volume statistics.
- Migrate a volume with back-end assistance.
- Retype a volume.

When you use back-end assisted volume migration, both source and destination clusters must be in the same HP LeftHand/StoreVirtual management group. The HP Left-

Hand/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-value pairs 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 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 an HP Lefthand/StoreVirtual SAN that supports the CLIQ interface. Every supported volume operation translates into a CLIQ call in the back end.

Warning

This driver has been deprecated as of the Liberty release, it will be removed in the M release. Please use the `HPLeftHandISCSIDriver` REST based driver instead.

Supported 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.

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.

Huawei storage driver

The Huawei driver supports the iSCSI and Fibre Channel connections and enables OceanStor T series V200R002, OceanStor 18000 series V100R001 and OceanStor V3 series V300R002 storage to provide block storage services for OpenStack.

Supported operations

- Create, delete, expand, attach, and detach volumes.
- Create and delete a snapshot.
- 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

1. Modify the `cinder.conf` configuration file and add `volume_driver` and `cinder_huawei_conf_file items`.

   - Example for configuring a storage system:

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

- Example for configuring multiple storage systems:

```
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
```

2. In `/etc/cinder`, create a driver configuration file. The driver configuration file name must be the same as the `cinder_huawei_conf_file` item in the `cinder_conf` configuration file.

3. **Configure product and protocol.**

   Product and Protocol indicate the storage system type and link type respectively. For the OceanStor 18000 series V100R001 storage systems, the driver configuration file is as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
    <Storage>
        <Product>18000</Product>
        <Protocol>iSCSI</Protocol>
        <RestURL>https://x.x.x.x/deviceManager/rest/</RestURL>
        <UserName>xxxxxxxx</UserName>
        <UserPassword>xxxxxxxx</UserPassword>
    </Storage>
    <LUN>
        <LUNType>Thick</LUNType>
        <WriteType>1</WriteType>
        <MirrorSwitch>0</MirrorSwitch>
        <LUNcopyWaitInterval>5</LUNcopyWaitInterval>
        <Timeout>432000</Timeout>
        <StoragePool>xxxxxxxx</StoragePool>
    </LUN>
    <iSCSI>
        <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
        <Initiator Name="xxxxxxxx" TargetIP="x.x.x.x"/>
        <Initiator Name="xxxxxxxx" TargetIP="x.x.x.x"/>
    </iSCSI>
    <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x"/>
</config>
```

Note

Note for fibre channel driver configuration

- In the configuration files of OceanStor T series V200R002 and OceanStor V3 V300R002, parameter configurations are the same with the excep-

tion of the RestURL parameter. The following describes how to configure the RestURL parameter:

```
<RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
```

- For a Fibre Channel driver, you do not need to configure an iSCSI target IP address. Delete the iSCSI configuration from the preceding examples.

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

This table describes the Huawei storage driver configuration options:

Table 2.8. Huawei storage driver configuration options

Property	Type	Default	Description
`Product`	Mandatory	-	Type of a storage product. Valid values are `T`, `TV3`, or `18000`.
`Protocol`	Mandatory	-	Type of a protocol. Valid values are `iSCSI` or `FC`.
`RestURL`	Mandatory	-	Access address of the Rest port (required only for the 18000)
`UserName`	Mandatory	-	User name of an administrator
`UserPassword`	Mandatory	-	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.

Property	Type	Default	Description
			Default value is 3, which indicates intelligent prefetch and is not required for the OceanStor 18000 series.
`Prefetch Value`	Optional	0	Cache prefetch value.
`LUNcopyWaitInterval`	Optional	5	After LUN copy is enabled, the plug-in frequently queries the copy progress. You can set a value to specify the query interval.
`Timeout`	Optional	432,000	Timeout period for waiting LUN copy of an array to complete.
`StoragePool`	Mandatory	-	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. Only one storage pool can be configured.

3. For details about LUN configuration information, see the **show lun general** command in the command-line interface (CLI) documentation or run the **help -c show lun general** 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.

4. Restart the Cinder service.

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 2.9. Description of GPFS storage configuration options

Configuration option = Default value	Description
[DEFAULT]	
`gpfs_images_dir` = *None*	(StrOpt) Specifies the path of the Image service repository in GPFS. Leave undefined if not storing images in GPFS.
`gpfs_images_share_mode` = *None*	(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.
`gpfs_max_clone_depth` = *0*	(IntOpt) Specifies an upper limit on the number of indirections required to reach a specific block due to snapshots

Configuration option = Default value	Description
	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.
gpfs_mount_point_base = *None*	(StrOpt) Specifies the path of the GPFS directory where Block Storage volume and snapshot files are stored.
gpfs_sparse_volumes = *True*	(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.
gpfs_storage_pool = *system*	(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 2.10. Volume Create Options for GPFS Volume Drive

Metadata Item Name	Description
fstype	Specifies whether to create a file system or a swap area on the new volume. If fstype=swap is specified, the mkswap command is used to create a swap area. Otherwise the mkfs command is passed the specified file system type, for example ext3, ext4 or ntfs.
fslabel	Sets the file system label for the file system specified by fstype option. This value is only used if fstype is specified.
data_pool_name	Specifies the GPFS storage pool to which the volume is to be assigned. Note: The GPFS storage pool must already have been created.
replicas	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 MaxDataReplicas attribute of the file system.
dio	Enables or disables the Direct I/O caching policy for the volume file. Valid values are yes and no.
write_affinity_depth	Specifies the allocation policy to be used for the volume file. Note: This option only works if allow-write-affinity is set for the GPFS data pool.
block_group_factor	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 allow-write-affinity is set for the GPFS data pool.

Metadata Item Name	Description
write_affinity_failure_group	Specifies the range of nodes (in GPFS shared nothing architecture) where replicas of blocks in the volume file are to be written. See *GPFS: Administration and Programming Reference* 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 2.11. List of configuration flags for Storwize storage and SVC driver

Flag name	Type	Default	Description
`san_ip`	Required		Management IP or host name
`san_ssh_port`	Optional	22	Management port
`san_login`	Required		Management login username
`san_password`	Required [a]		Management login password
`san_private_key`	Required [a]		Management login SSH private key
`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]

[a] The 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.

[b] The 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.

[c] Defines 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.

[d] Defines 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.

[e]Defines 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.

[f]The 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).

[g]Multipath for iSCSI connections requires no storage-side configuration and is enabled if the compute host has multipath configured.

[h]This 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 2.12. Description of IBM Storwise driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
storwize_svc_allow_tenant_qos = *False*	(BoolOpt) Allow tenants to specify QOS on create
storwize_svc_connection_protocol = *iSCSI*	(StrOpt) Connection protocol (iSCSI/FC)
storwize_svc_flashcopy_timeout = *120*	(IntOpt) Maximum number of seconds to wait for FlashCopy to be prepared. Maximum value is 600 seconds (10 minutes)
storwize_svc_iscsi_chap_enabled = *True*	(BoolOpt) Configure CHAP authentication for iSCSI connections (Default: Enabled)
storwize_svc_multihostmap_enabled = *True*	(BoolOpt) Allows vdisk to multi host mapping
storwize_svc_multipath_enabled = *False*	(BoolOpt) Connect with multipath (FC only; iSCSI multipath is controlled by Nova)
storwize_svc_npiv_compatibility_mode = *False*	(BoolOpt) Indicate whether svc driver is compatible for NPIV setup. If it is compatible, it will allow no wwpns being returned on get_conn_fc_wwpns during initialize_connection
storwize_svc_stretched_cluster_partner = *None*	(StrOpt) If operating in stretched cluster mode, specify the name of the pool in which mirrored copies are stored.Example: "pool2"
storwize_svc_vol_autoexpand = *True*	(BoolOpt) Storage system autoexpand parameter for volumes (True/False)
storwize_svc_vol_compression = *False*	(BoolOpt) Storage system compression option for volumes
storwize_svc_vol_easytier = *True*	(BoolOpt) Enable Easy Tier for volumes
storwize_svc_vol_grainsize = *256*	(IntOpt) Storage system grain size parameter for volumes (32/64/128/256)
storwize_svc_vol_iogrp = *0*	(IntOpt) The I/O group in which to allocate volumes
storwize_svc_vol_rsize = *2*	(IntOpt) Storage system space-efficiency parameter for volumes (percentage)
storwize_svc_vol_warning = *0*	(IntOpt) Storage system threshold for volume capacity warnings (percentage)
storwize_svc_volpool_name = *volpool*	(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 Stor-

wize/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)

QOS

The Storwize driver provides QOS support for storage volumes by controlling the I/O amount. QOS is enabled by editing the `etc/cinder/cinder.conf` file and setting the `storwize_svc_allow_tenant_qos` to *True*.

There are three ways to set the Storwize `IOThrotting` parameter for storage volumes:

- Add the `qos:IOThrottling` key into a QOS specification and associate it with a volume type.
- Add the `qos:IOThrottling` key into an extra specification with a volume type.
- Add the `qos:IOThrottling` key to the storage volume metadata.

Note

If you are changing a volume type with QOS to a new volume type without QOS, the QOS configuration settings will be removed.

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:

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

Note

When you change the `rsize`, `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.

IBM XIV and DS8000 volume driver

The IBM Storage Driver for OpenStack is a Block Storage driver that supports IBM XIV and IBM DS8000 storage systems over Fiber channel and iSCSI.

Set the following in your `cinder.conf`, and use the following options to configure it.

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

Table 2.13. Description of IBM XIV and DS8000 volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`san_clustername` =	(StrOpt) Cluster name to use for creating volumes
`san_ip` =	(StrOpt) IP address of SAN controller

Configuration option = Default value	Description
`san_login = admin`	(StrOpt) Username for SAN controller
`san_password =`	(StrOpt) Password for SAN controller
`xiv_chap = disabled`	(StrOpt) CHAP authentication mode, effective only for iscsi (disabled|enabled)
`xiv_ds8k_connection_type = iscsi`	(StrOpt) Connection type to the IBM Storage Array
`xiv_ds8k_proxy = xiv_ds8k_openstack.nova_proxy.XIVDS8KNovaProxy`	(StrOpt) Proxy driver that connects to the IBM Storage Array

Note

To use the IBM Storage Driver for OpenStack you must download and install the package available at: http://www.ibm.com/support/fixcentral/swg/selectFixes?parent=Enterprise%2BStorage%2BServers&product=ibm/Storage_Disk/XIV+Storage+System+%282810,+2812%29&release=All&platform=All&function=all

For full documentation refer to IBM's online documentation available at http://pic.dhe.ibm.com/infocenter/strhosts/ic/topic/com.ibm.help.strghosts.doc/nova-homepage.html.

IBM FlashSystem volume driver

The volume driver for FlashSystem provides OpenStack Block Storage hosts with access to IBM FlashSystems.

Configure FlashSystem

Data Path configuration

The FlashSystem should be configured for Fiber Channel (FC).

Using Fiber Channel (FC), each FlashSystem node should have at least one WWPN port configured. If the `flashsystem_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 section "Enable IBM FlashSystem FC driver"). 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 to the driver.

Note

Using FC, ensure that the block storage hosts have FC connectivity to the FlashSystem.

Configure storage array

The volume driver requires a pre-defined array. Users need to create an array on FlashSystem before using the volume driver. An existing array also can be used and existing data will not be destroyed.

Note

FlashSystem can only create one array, so no configuration option is needed for the IBM FlashSystem driver to assign it.

Configure user authentication for the driver

The driver requires access to the FlashSystem management interface. The driver communicates with the management using SSH. The driver should be provided with the FlashSystem 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` driver has SSH network access to the storage system.

Using password authentication, assign a password to the user on the FlashSystem. See the driver configuration flags for the user and password in section "Enable IBM FlashSystem FC driver".

IBM FlashSystem FC driver

Enable IBM FlashSystem FC driver

Set the volume driver to the FlashSystem driver by setting the `volume_driver` option in configuration file `cinder.conf` as follows:

```
volume_driver = cinder.volume.drivers.ibm.flashsystem.FlashSystemDriver
```

To enable IBM FlashSystem FC driver, configure the following options in configuration file `cinder.conf`:

Table 2.14. List of configuration flags for IBM FlashSystem FC driver

Flag name	Type	Default	Description
`san_ip`	Required		Management IP or host name
`san_ssh_port`	Optional	22	Management port
`san_login`	Required		Management login user name
`san_password`	Required		Management login password
`flashsystem_connection_protocol`	Optional	FC	Connection protocol should be FC
`flashsystem_multipath_enabled`	Optional	False	Enable multipath for FC connections
`flashsystem_multihost_enabled`	Optional	True	Enable mapping vdisks to multiple hosts [a]

[a] This 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.

Limitation and known issues

- IBM FlashSystem cinder driver only supports Fibre Channel (FC) now.

Supported operations

These operations are supported:

- 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.
- Extend a volume.
- Get volume statistics.

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` configuration file, and use the following options to configure for iSCSI transport:

```
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
iscsi_protocol = iscsi
```

Use the following options to configure for the iSER transport:

```
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
iscsi_protocol = iser
```

Table 2.15. Description of LVM configuration options

Configuration option = Default value	Description
[DEFAULT]	
`lvm_conf_file` = */etc/cinder/lvm.conf*	(StrOpt) LVM conf file to use for the LVM driver in Cinder; this setting is ignored if the specified file does not exist (You can also specify 'None' to not use a conf file even if one exists).
`lvm_mirrors` = *0*	(IntOpt) If >0, create LVs with multiple mirrors. Note that this requires lvm_mirrors + 2 PVs with available space
`lvm_type` = *default*	(StrOpt) Type of LVM volumes to deploy
`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.

Note

With the Juno release of OpenStack, OpenStack Block Storage has introduced the concept of "storage pools", in which a single OpenStack Block Storage back end may present one or more logical storage resource pools from which OpenStack Block Storage will select as a storage location when provisioning volumes.

In releases prior to Juno, the NetApp unified driver contained some "scheduling" logic that determined which NetApp storage container (namely, a FlexVol volume for Data ONTAP, or a dynamic disk pool for E-Series) that a new OpenStack Block Storage volume would be placed into.

With the introduction of pools, all scheduling logic is performed completely within the OpenStack Block Storage scheduler, as each NetApp storage container is directly exposed to the OpenStack Block Storage scheduler as a storage pool; whereas previously, the NetApp unified driver presented an aggregated view to the scheduler and made a final placement decision as to which NetApp storage container the OpenStack Block Storage volume would be provisioned into.

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 = port
netapp_login = username
netapp_password = password
```

Note

To use the iSCSI protocol, you must override the default value of `netapp_storage_protocol` with `iscsi`.

Table 2.16. Description of NetApp cDOT iSCSI driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`netapp_login` = *None*	(StrOpt) Administrative user account name used to access the storage system or proxy server.
`netapp_partner_backend_name` = *None*	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
`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` = *None*	(IntOpt) The TCP port to use for communication with the storage system or proxy server. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS; E-Series will use 8080 for HTTP and 8443 for HTTPS.
`netapp_size_multiplier` = *1.2*	(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.
`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.
`netapp_transport_type` = *http*	(StrOpt) The transport protocol used when communicating with the storage system or proxy server.
`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.

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 NetApp OpenStack Deployment and Operations Guide.

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 = port
netapp_login = username
netapp_password = password
nfs_shares_config = /etc/cinder/nfs_shares
```

Table 2.17. Description of NetApp cDOT NFS driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`expiry_thres_minutes` = *720*	(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.
`netapp_copyoffload_tool_path` = *None*	(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 cinder-volume process to execute the file.
`netapp_login` = *None*	(StrOpt) Administrative user account name used to access the storage system or proxy server.
`netapp_partner_backend_name` = *None*	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
`netapp_password` = *None*	(StrOpt) Password for the administrative user account specified in the netapp_login option.

Configuration option = Default value	Description
netapp_server_hostname = *None*	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
netapp_server_port = *None*	(IntOpt) The TCP port to use for communication with the storage system or proxy server. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS; E-Series will use 8080 for HTTP and 8443 for HTTPS.
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.
netapp_transport_type = *http*	(StrOpt) The transport protocol used when communicating with the storage system or proxy server.
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.
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 2.23, "Description of NFS storage configuration options" [94].

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 NetApp OpenStack Deployment and Operations Guide.

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

Extra spec	Type	Description
`netapp_raid_type`	String	Limit the candidate volume list based on one of the following raid types: `raid4, raid_dp`.
`netapp_disk_type`	String	Limit the candidate volume list based on one of the following disk types: `ATA, BSAS, EATA, FCAL, FSAS, LUN, MSATA, SAS, SATA, SCSI, XATA, XSAS, or SSD`.
`netapp:qos_policy_group`[a]	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.
`netapp_mirrored`	Boolean	Limit the candidate volume list to only the ones that are mirrored on the storage controller.
`netapp_unmirrored`[b]	Boolean	Limit the candidate volume list to only the ones that are not mirrored on the storage controller.
`netapp_dedup`	Boolean	Limit the candidate volume list to only the ones that have deduplication enabled on the storage controller.
`netapp_nodedup`[b]	Boolean	Limit the candidate volume list to only the ones that have deduplication disabled on the storage controller.
`netapp_compression`	Boolean	Limit the candidate volume list to only the ones that have compression enabled on the storage controller.
`netapp_nocompression`[b]	Boolean	Limit the candidate volume list to only the ones that have compression disabled on the storage controller.
`netapp_thin_provisioned`	Boolean	Limit the candidate volume list to only the ones that support thin provisioning on the storage controller.
`netapp_thick_provisioned`[b]	Boolean	Limit the candidate volume list to only the ones that support thick provisioning on the storage controller.

[a] Please note that this extra spec has a colon (`:`) in its name because it is used by the driver to assign the QoS policy group to the OpenStack Block Storage volume after it has been provisioned.

[b] In the Juno release, these negative-assertion extra specs are formally deprecated by the NetApp unified driver. Instead of using the deprecated negative-assertion extra specs (for example, `netapp_unmirrored`) with a value of `true`, use the corresponding positive-assertion extra spec (for example, `netapp_mirrored`) with 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

To use the iSCSI protocol, you must override the default value of `netapp_storage_protocol` with `iscsi`.

Table 2.19. Description of NetApp 7-Mode iSCSI driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`netapp_login` = *None*	(StrOpt) Administrative user account name used to access the storage system or proxy server.
`netapp_partner_backend_name` = *None*	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
`netapp_password` = *None*	(StrOpt) Password for the administrative user account specified in the netapp_login option.

Configuration option = Default value	Description
netapp_server_hostname = *None*	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
netapp_server_port = *None*	(IntOpt) The TCP port to use for communication with the storage system or proxy server. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS; E-Series will use 8080 for HTTP and 8443 for HTTPS.
netapp_size_multiplier = *1.2*	(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.
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.
netapp_transport_type = *http*	(StrOpt) The transport protocol used when communicating with the storage system or proxy server.
netapp_vfiler = *None*	(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. Only use this option when utilizing the MultiStore feature on the NetApp storage system.
netapp_volume_list = *None*	(StrOpt) This option is only utilized when the storage protocol is configured to use iSCSI or FC. 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 NetApp OpenStack Deployment and Operations Guide.

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 2.20. Description of NetApp 7-Mode NFS driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
expiry_thres_minutes = *720*	(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.
netapp_login = *None*	(StrOpt) Administrative user account name used to access the storage system or proxy server.
netapp_partner_backend_name = *None*	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
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 = *None*	(IntOpt) The TCP port to use for communication with the storage system or proxy server. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS; E-Series will use 8080 for HTTP and 8443 for HTTPS.
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.
netapp_transport_type = *http*	(StrOpt) The transport protocol used when communicating with the storage system or proxy server.
netapp_vfiler = *None*	(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. Only use this option when utilizing the MultiStore feature on the NetApp storage system.
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. For a description of these, see Table 2.23, "Description of NFS storage configuration options" [94].

Tip

For more information on these options and other deployment and operational scenarios, visit the NetApp OpenStack Deployment and Operations Guide.

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.

The use of multipath and DM-MP are required when using the OpenStack Block Storage driver for E-Series. In order for OpenStack Block Storage and OpenStack Compute to take advantage of multiple paths, the following configuration options must be correctly configured:

- The `use_multipath_for_image_xfer` option should be set to `True` in the `cinder.conf` file within the driver-specific stanza (for example, `[myDriver]`).
- The `iscsi_use_multipath` option should be set to `True` in the `nova.conf` file within the `[libvirt]` stanza.

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
use_multipath_for_image_xfer = True
```

Note

To use the E-Series driver, you must override the default value of `netapp_storage_family` with `eseries`.

Note

To use the iSCSI protocol, you must override the default value of `netapp_storage_protocol` with `iscsi`.

Table 2.21. Description of NetApp E-Series driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`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_eseries_host_type` = *linux_dm_mp*	(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.
`netapp_login` = *None*	(StrOpt) Administrative user account name used to access the storage system or proxy server.
`netapp_partner_backend_name` = *None*	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
`netapp_password` = *None*	(StrOpt) Password for the administrative user account specified in the netapp_login option.
`netapp_sa_password` = *None*	(StrOpt) Password for the NetApp E-Series storage array.
`netapp_server_hostname` = *None*	(StrOpt) The hostname (or IP address) for the storage system or proxy server.
`netapp_server_port` = *None*	(IntOpt) The TCP port to use for communication with the storage system or proxy server. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS; E-Series will use 8080 for HTTP and 8443 for HTTPS.
`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_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

Configuration option = Default value	Description
	comma separated list of disk pool names to be used for provisioning.
netapp_transport_type = *http*	(StrOpt) The transport protocol used when communicating with the storage system or proxy server.
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 NetApp OpenStack Deployment and Operations Guide.

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 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.common.NetAppDriver
   netapp_storage_family = ontap_cluster
   netapp_storage_protocol = nfs
   ```

3. NetApp iSCSI direct driver for Data ONTAP operating in 7-Mode storage controller in Grizzly (or earlier)

   ```
   volume_driver = cinder.volume.drivers.netapp.iscsi.
   NetAppDirect7modeISCSIDriver
   ```

NetApp unified driver configuration

```
volume_driver = cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family = ontap_7mode
netapp_storage_protocol = iscsi
```

4. NetApp NFS direct driver for Data ONTAP operating in 7-Mode storage controller in Grizzly (or earlier)

```
volume_driver = cinder.volume.drivers.netapp.nfs.NetAppDirect7modeNfsDriver
```

NetApp unified driver configuration

```
volume_driver = cinder.volume.drivers.netapp.common.NetAppDriver
netapp_storage_family = ontap_7mode
netapp_storage_protocol = nfs
```

Deprecated NetApp drivers

This section lists the NetApp drivers in earlier releases that are deprecated in Havana.

1. NetApp iSCSI driver for clustered Data ONTAP.

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

2. NetApp NFS driver for clustered Data ONTAP.

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

3. NetApp iSCSI driver for Data ONTAP operating in 7-Mode storage controller.

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

4. NetApp NFS driver for Data ONTAP operating in 7-Mode storage controller.

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

Note

For support information on deprecated NetApp drivers in the Havana release, visit the NetApp OpenStack Deployment and Operations Guide.

Nimble volume driver

Set the following in your `cinder.conf` to use the Nimble volume driver:

```
volume_driver = cinder.volume.drivers.nimble.NimbleISCSIDriver
```

Use the following options to configure the volume driver:

Table 2.22. Description of Nimble driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`nimble_pool_name` = *`default`*	(StrOpt) Nimble Controller pool name
`nimble_subnet_label` = *	(StrOpt) Nimble Subnet Label

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 2.23. Description of NFS storage configuration options

Configuration option = Default value	Description
[DEFAULT]	
`nfs_mount_attempts` = *3*	(IntOpt) The number of attempts to mount nfs shares before raising an error. At least one attempt will be made to mount an nfs share, regardless of the value specified.
`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) Base dir containing mount points for nfs shares.
`nfs_oversub_ratio` = *1.0*	(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.
`nfs_shares_config` = */etc/cinder/nfs_shares*	(StrOpt) File with the list of available nfs shares
`nfs_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.
`nfs_used_ratio` = *0.95*	(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 be-

havior 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.

ProphetStor Fibre Channel and iSCSI drivers

ProhetStor Fibre Channel and iSCSI drivers add support for ProphetStor Flexvisor through OpenStack Block Storage. ProphetStor Flexvisor enables commodity x86 hardware as software-defined storage leveraging well-proven ZFS for disk management to provide enterprise grade storage services such as snapshots, data protection with different RAID levels, replication, and deduplication.

The `DPLFCDriver` and `DPLISCSIDriver` drivers run volume operations by communicating with the ProphetStor storage system over HTTPS.

Supported 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.
- Extend a volume.

Enable the Fibre Channel or iSCSI drivers

The `DPLFCDriver` and `DPLISCSIDriver` are installed with the OpenStack software.

1. Query storage pool id for configure `dpl_pool` of the `cinder.conf`.
   a. Logon onto the storage system with administrator access.

```
$ ssh root@STORAGE IP ADDRESS
```

b. View the current usable pool id.

```
$ flvcli show pool list
- d5bd40b58ea84e9da09dcf25a01fdc07 : default_pool_dc07
```

c. Use `d5bd40b58ea84e9da09dcf25a01fdc07` to config the `dpl_pool` of `/etc/cinder/cinder.conf`.

Note

Other management command can reference by command help **flvcli -h**.

2. Make the following changes on the volume node `/etc/cinder/cinder.conf` file.

```
# IP address of SAN controller (string value)
san_ip=STORAGE IP ADDRESS

# Username for SAN controller (string value)
san_login=USERNAME

# Password for SAN controller (string value)
san_password=PASSWORD

# Use thin provisioning for SAN volumes? (boolean value)
san_thin_provision=true

# The port that the iSCSI daemon is listening on. (integer value)
iscsi_port=3260

# DPL pool uuid in which DPL volumes are stored. (string value)
dpl_pool=d5bd40b58ea84e9da09dcf25a01fdc07

# DPL port number. (integer value)
dpl_port=8357

# Uncomment one of the next two option to enable Fibre channel or iSCSI
# FIBRE CHANNEL(uncomment the next line to enable the FC driver)
#volume_driver=cinder.volume.drivers.prophetstor.dpl_fc.DPLFCDriver
# iSCSI (uncomment the next line to enable the iSCSI driver)
#volume_driver=cinder.volume.drivers.prophetstor.dpl_iscsi.DPLISCSIDriver
```

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

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

The following table contains the options supported by the ProphetStor storage driver.

Table 2.24. Description of ProphetStor Fibre Channel and iSCSi drivers configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
dpl_pool =	(StrOpt) DPL pool uuid in which DPL volumes are stored.
dpl_port = *8357*	(IntOpt) DPL port number.
iscsi_port = *3260*	(IntOpt) The port that the iSCSI daemon is listening on
san_ip =	(StrOpt) IP address of SAN controller
san_login = *admin*	(StrOpt) Username for SAN controller
san_password =	(StrOpt) Password for SAN controller
san_thin_provision = *True*	(BoolOpt) Use thin provisioning for SAN volumes?

Pure Storage volume driver

The Pure Storage FlashArray volume driver for OpenStack Block Storage interacts with configured Pure Storage arrays and supports various operations.

This driver can be configured in OpenStack Block Storage to work with the iSCSI storage protocol.

This driver is compatible with Purity FlashArrays that support the REST API (Purity 3.4.0 and newer) and that are capable of iSCSI connectivity. This release supports installation with OpenStack clusters running the Kilo version that use the KVM or QEMU hypervisors together with OpenStack Compute service's libvirt driver.

Limitations and known issues

If you do not set up the nodes hosting instances to use multipathing, all iSCSI connectivity will use a single physical 10-gigabit Ethernet port on the array. In addition to significantly limiting the available bandwidth, this means you do not have the high-availability and non-disruptive upgrade benefits provided by FlashArray.

Workaround: You must set up multipathing on your hosts.

Supported operations

- Create, delete, attach, detach, retype, clone, and extend volumes.
- Create a volume from snapshot.
- Create, list, and delete volume snapshots.
- Create, list, update, and delete consistency groups.
- Create, list, and delete consistency group snapshots.
- Manage and unmanage a volume.
- Get volume statistics.
- Create a thin provisioned volume.

Configure OpenStack and Purity

You need to configure both your Purity array and your OpenStack cluster.

Note

These instructions assume that the `cinder-api` and `cinder-scheduler` services are installed and configured in your OpenStack cluster.

- **Configure the OpenStack Block Storage service**

  In these steps, you will edit the `cinder.conf` file to configure OpenStack Block Storage service to enable multipathing and to use the Pure Storage FlashArray as back-end storage.

  a. **Retrieve an API token from Purity**

  The OpenStack Block Storage service configuration requires an API token from Purity. Actions performed by the volume driver use this token for authorization. Also, Purity logs the volume driver's actions as being performed by the user who owns this API token.

  If you created a Purity user account that is dedicated to managing your OpenStack Block Storage volumes, copy the API token from that user account.

  Use the appropriate create or list command below to display and copy the Purity API token:

  - To create a new API token:

    ```
    $ pureadmin create --api-token USER
    ```

    The following is an example output:

    ```
    $ pureadmin create --api-token pureuser
    Name      API Token                             Created
    pureuser  902fdca3-7e3f-d2e4-d6a6-24c2285fe1d9  2014-08-04
     14:50:30
    ```

  - To list an existing API token:

    ```
    $ pureadmin list --api-token --expose USER
    ```

    The following is an example output:

    ```
    $ pureadmin list --api-token --expose pureuser
    Name      API Token                             Created
    pureuser  902fdca3-7e3f-d2e4-d6a6-24c2285fe1d9  2014-08-04
     14:50:30
    ```

  b. Copy the API token retrieved (`902fdca3-7e3f-d2e4-d6a6-24c2285fe1d9` from the examples above) to use in the next step.

  c. **Edit the OpenStack Block Storage service configuration file**

  The following sample `/etc/cinder/cinder.conf` configuration lists the relevant settings for a typical Block Storage service using a single Pure Storage array:

```
[DEFAULT]
....
enabled_backends = puredriver-1
default_volume_type = puredriver-1
....

[puredriver-1]
volume_backend_name = puredriver-1
volume_driver = cinder.volume.drivers.pure.PureISCSIDriver
san_ip = IP_PURE_MGMT
pure_api_token = PURE_API_TOKEN
use_multipath_for_image_xfer = True
```

Replace the following variables accordingly:

IP_PURE_MGMT The IP address of the Pure Storage array's management interface or a domain name that resolves to that IP address.

PURE_API_TOKEN The Purity Authorization token that the volume driver uses to perform volume management on the Pure Storage array.

Note

The volume driver automatically creates purity host objects for initiators as needed. If CHAP authentication is enabled via the `use_chap_auth` settting, you must ensure there are no manually created host objects with IQN's that will be used by the OpenStack Block Storage. The driver will only modify credentials on hosts that it manages.

Quobyte driver

Quobyte provides a software defined storage system running on plain Linux servers. Quobyte allows reconfiguration at runtime without storage service interruption. Storage behaviour is fully configurable via policies which also control automation and hardware independent configuration. Quobyte users can find more information via the documentation link at http://support.quobyte.com/.

This driver enables the use of Quobyte storage in a similar fashion as NFS. It supports basic volume operations, including snapshot and clone.

To use Block Storage with Quobyte, first set the `volume_driver` configuration option in the `cinder.conf` file:

```
volume_driver = cinder.volume.drivers.quobyte.QuobyteDriver
```

The following table contains the configuration options supported by the Quobyte volume driver.

Table 2.25. Description of Quobyte USP volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
quobyte_client_cfg = *None*	(StrOpt) Path to a Quobyte Client configuration file.
quobyte_mount_point_base = *$state_path/mnt*	(StrOpt) Base dir containing the mount point for the Quobyte volume.
quobyte_qcow2_volumes = *True*	(BoolOpt) Create volumes as QCOW2 files rather than raw files.
quobyte_sparsed_volumes = *True*	(BoolOpt) Create volumes as sparse 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.
quobyte_volume_url = *None*	(StrOpt) URL to the Quobyte volume e.g., quobyte://<DIR host>/<volume name>

Scality SOFS driver

The Scality SOFS volume driver interacts with configured sfused mounts.

The Scality SOFS driver manages volumes as sparse files stored on a Scality Ring through sfused. Ring connection settings and sfused options are defined in the `cinder.conf` file and the configuration file pointed to by the `scality_sofs_config` option, typically `/etc/sfused.conf`.

Supported operations

The Scality SOFS volume driver provides the following Block Storage volume operations:

- Create, delete, attach (map), and detach (unmap) 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.
- Extend a volume.
- Backup a volume.
- Restore backup to new or existing volume.

Configuration

Use the following instructions to update the `cinder.conf` configuration file:

Example 2.5. Sample Block Storage Configuration

```
[DEFAULT]
enabled_backends = scality-1

[scality-1]
volume_driver=cinder.volume.drivers.scality.ScalityDriver
volume_backend_name = scality-1

scality_sofs_config=/etc/sfused.conf
scality_sofs_mount_point=/cinder
scality_sofs_volume_dir=cinder/volumes
```

Use the following instructions to update the `nova.conf` configuration file:

Example 2.6. Sample Compute Configuration

```
[libvirt]
scality_sofs_mount_point = /cinder
scality_sofs_config = /etc/sfused.conf
```

Table 2.26. Description of Scality SOFS volume driver configuration options

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
`scality_sofs_volume_dir` = *cinder/volumes*	(StrOpt) Path from Scality SOFS root to volume dir

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
```

SambaFS driver

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

Note

The SambaFS driver requires **qemu-img** version 1.7 or higher on Linux nodes, and **qemu-img>** version 1.6 or higher on Windows nodes.

```
volume_driver=cinder.volume.drivers.smbfs.SmbfsDriver
```

Table 2.27. Description of Samba volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
smbfs_default_volume_format = *qcow2*	(StrOpt) Default format that will be used when creating volumes if no volume format is specified.
smbfs_mount_options = *noperm,file_mode=0775,dir_mode=0775*	(StrOpt) Mount options passed to the smbfs client. See mount.cifs man page for details.
smbfs_mount_point_base = *$state_path/mnt*	(StrOpt) Base dir containing mount points for smbfs shares.
smbfs_oversub_ratio = *1.0*	(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.
smbfs_shares_config = */etc/cinder/smbfs_shares*	(StrOpt) File with the list of available smbfs shares.
smbfs_sparsed_volumes = *True*	(BoolOpt) Create volumes as sparsed files which take no space rather than regular files when using raw format, in which case volume creation takes lot of time.
smbfs_used_ratio = *0.95*	(FloatOpt) Percent of ACTUAL usage of the underlying volume before no new volumes can be allocated to the volume destination.

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 = sfpassword     # 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 2.28. Description of SolidFire driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`sf_account_prefix` = *None*	(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 hostsname (previous default behavior). The default is NO prefix.
`sf_allow_template_caching` = *True*	(BoolOpt) Create an internal cache of copy of images when a bootable volume is created to eliminate fetch from glance and qemu-conversion on subsequent calls.
`sf_allow_tenant_qos` = *False*	(BoolOpt) Allow tenants to specify QOS on create
`sf_api_port` = *443*	(IntOpt) SolidFire API port. Useful if the device api is behind a proxy on a different port.
`sf_emulate_512` = *True*	(BoolOpt) Set 512 byte emulation on volume creation;
`sf_template_account_name` = *openstack-vtemplate*	(StrOpt) Account name on the SolidFire Cluster to use as owner of template/cache volumes (created if does not exist).

Tintri

Tintri VMstore is a smart storage that sees, learns and adapts for cloud and virtualization. Tintri Cinder driver will interact with configured VMstore running Tintri OS 4.0 and above. It supports various operations using Tintri REST APIs and NFS protocol.

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

```
volume_driver = cinder.volume.drivers.tintri.TintriDriver
# Mount options passed to the nfs client. See section of the
# nfs man page for details. (string value)
nfs_mount_options=vers=3,lookupcache=none

#
# Options defined in cinder.volume.drivers.tintri
#

# The hostname (or IP address) for the storage system (string
# value)
tintri_server_hostname={Tintri VMstore Management IP}

# User name for the storage system (string value)
tintri_server_username={username}

# Password for the storage system (string value)
tintri_server_password={password}
```

```
# API version for the storage system (string value)
#tintri_api_version=v310

# Following options needed for NFS configuration
# File with the list of available nfs shares (string value)
#nfs_shares_config=/etc/cinder/nfs_shares
```

Following changes are needed in /etc/cinder/nfs_shares

```
# Edit /etc/cinder/nfs_shares to add one or more Tintri VMstore mount points
# associated with already configured VMstore management IP in cinder.conf
{vmstore_data_ip}:/tintri/{submount1}
{vmstore_data_ip}:/tintri/{submount2}
```

Following changes are needed in `nova.conf` file:

```
# Edit /etc/nova/nova.conf
nfs_mount_options=vers=3
```

Table 2.29. Description of Tintri driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`tintri_server_hostname =` *None*	(StrOpt) Tintri VMstore management IP address
`tintri_server_username =` *None*	(StrOpt) Tintri VMstore username
`tintri_server_password =` *None*	(StrOpt) Tintri VMstore password for the user
`tintri_api_version =` *v310*	(StrOpt) Tintri VMstore API version

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. The

reason for this requirement is that data stores visible to the instance determine where to place the volume. Before the service creates the VMDK file, attach a volume to the target instance.

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.

With the update to ESX version 6.0, the VMDK driver now supports NFS version 4.1.

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 2.30. Description of VMware configuration options

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` = *0.5*	(FloatOpt) The interval (in seconds) for polling remote tasks invoked on VMware ESX/VC server.
`vmware_tmp_dir` = */tmp*	(StrOpt) Directory where virtual disks are stored during volume backup and restore.
`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 file types `thin`, `lazyZeroedThick` (sometimes called thick or flat), or `eagerZeroedThick`.

A thin virtual disk is allocated and zeroed on demand as the space is used. Unused space on a Thin disk is available to other users.

A lazy zeroed thick virtual disk will have all space allocated at disk creation. This reserves the entire disk space, so it is not available to other users at any time.

An eager zeroed thick virtual disk is similar to a lazy zeroed thick disk, in that the entire disk is allocated at creation. However, in this type, any previous data will be wiped clean on the disk before the write. This can mean that the disk will take longer to create, but can also prevent issues with stale data on physical media.

Use the `vmware:vmdk_type` extra spec key with the appropriate value to specify the VMDK disk file type. This table shows the mapping between the extra spec entry and the VMDK disk file type:

Table 2.31. Extra spec entry to VMDK disk file type mapping

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

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

The following example shows how to create a `lazyZeroedThick` 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 2.32. Extra spec entry to clone type mapping

Clone type	Extra spec key	Extra spec value
full	vmware:clone_type	full

Clone type	Extra spec key	Extra spec value
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 a source volume, which is created from an image:

```
$ cinder type-create fast_clone
$ cinder type-key fast_clone set vmware:clone_type=linked
$ cinder create --image-id 9cb87f4f-a046-47f5-9b7c-d9487b3c7cd4 --volume-type
 fast_clone --display-name source-vol 1
$ cinder create --source-volid 25743b9d-3605-462b-b9eb-71459fe2bb35 --display-
name dest-vol 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 2.2. 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” [110].

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, delete, attach, and detach volumes.

Note

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.

- Create, list, and delete volume snapshots.

Note

Allowed only if volume is not attached to an instance.

- Create a volume from a snapshot.
- Copy an image to a volume.

Note

Only images in `vmdk` disk format with `bare` container format are supported. The `vmware_disktype` property of the image can be `preallocated`, `sparse`, `streamOptimized` or `thin`.

- Copy a volume to an image.

Note

- Allowed only if the volume is not attached to an instance.

- This operation creates a `streamOptimized` disk image.

- Clone a volume.

Note

Supported only if the source volume is not attached to an instance.

- Backup a volume.

Note

This operation creates a backup of the volume in `streamOptimized` disk format.

- Restore backup to new or existing volume.

Note

Supported only if the existing volume doesn't contain snapshots.

- Change the type of a volume.

Note

This operation is supported only if the volume state is `available`.

- Extend a volume.

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 2.3. 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

For details about creating storage policies 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-freespace/totalspace)` meters.

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 iSCSI volume driver

Windows Server 2012 and Windows Storage Server 2012 offer an integrated iSCSI Target service that can be used with OpenStack Block Storage in your stack. Being entirely a software solution, consider it in particular for mid-sized networks where the costs of a SAN might be excessive.

The Windows `cinder-volume` driver works with OpenStack Compute on any hypervisor. It includes snapshotting support and the "boot from volume" feature.

This driver creates volumes backed by fixed-type VHD images on Windows Server 2012 and dynamic-type VHDX on Windows Server 2012 R2, stored locally on a user-specified path. The system uses those images as iSCSI disks and exports them through iSCSI targets. Each volume has its own iSCSI target.

This driver has been tested with Windows Server 2012 and Windows Server R2 using the Server and Storage Server distributions.

Install the `cinder-volume` service as well as the required Python components directly onto the Windows node.

You may install and configure `cinder-volume` and its dependencies manually using the following guide or you may use the `Cinder Volume Installer`, presented below.

Installing using the OpenStack cinder volume installer

In case you want to avoid all the manual setup, you can use Cloudbase Solutions' installer. You can find it at https://www.cloudbase.it/downloads/CinderVolumeSetup_Beta.msi. It installs an independent Python environment, in order to avoid conflicts with existing applications, dynamically generates a `cinder.conf` file based on the parameters provided by you.

`cinder-volume` will be configured to run as a Windows Service, which can be restarted using:

```
PS C:\> net stop cinder-volume ; net start cinder-volume
```

The installer can also be used in unattended mode. More details about how to use the installer and its features can be found at https://www.cloudbase.it

Windows Server configuration

The required service in order to run `cinder-volume` on Windows is `wintarget`. This will require the iSCSI Target Server Windows feature to be installed. You can install it by running the following command:

```
PS C:\> Add-WindowsFeature
FS-iSCSITarget-ServerAdd-WindowsFeatureFS-iSCSITarget-Server
```

Note

The Windows Server installation requires at least 16 GB of disk space. The volumes hosted by this node need the extra space.

For `cinder-volume` to work properly, you must configure NTP as explained in the section called “Configure NTP” [250].

Next, install the requirements as described in the section called “Requirements” [253].

Getting the code

Git can be used to download the necessary source code. The installer to run Git on Windows can be downloaded here:

https://github.com/msysgit/msysgit/releases/download/Git-1.9.2-preview20140411/Git-1.9.2-preview20140411.exe

Once installed, run the following to clone the OpenStack Block Storage code.

```
PS C:\> git.exe clone https://github.com/openstack/cinder.git
```

Configure cinder-volume

The `cinder.conf` file may be placed in `C:\etc\cinder`. Below is a config sample for using the Windows iSCSI Driver:

```
[DEFAULT]
auth_strategy = keystone
volume_name_template = volume-%s
volume_driver = cinder.volume.drivers.windows.WindowsDriver
glance_api_servers = IP_ADDRESS:9292
rabbit_host = IP_ADDRESS
rabbit_port = 5672
sql_connection = mysql://root:Passw0rd@IP_ADDRESS/cinder
windows_iscsi_lun_path = C:\iSCSIVirtualDisks
verbose = True
rabbit_password = Passw0rd
logdir = C:\OpenStack\Log\
image_conversion_dir = C:\ImageConversionDir
debug = True
```

The following table contains a reference to the only driver specific option that will be used by the Block Storage Windows driver:

Table 2.33. Description of Windows configuration options

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

Running cinder-volume

After configuring `cinder-volume` using the `cinder.conf` file, you may use the following commands to install and run the service (note that you must replace the variables with the proper paths):

```
PS C:\> python $CinderClonePath\setup.py install
PS C:\> cmd /c C:\python27\python.exe c:\python27\Scripts\cinder-volume" --
config-file $CinderConfPath
```

X-IO volume driver

The X-IO volume driver for OpenStack Block Storage enables ISE products to be managed by OpenStack Block Storage nodes. This driver can be configured to work with iSCSI and Fibre Channel storage protocols. The X-IO volume driver allows the cloud operator to take advantage of ISE features like Quality of Service and Continous Adaptive Data Placement (CADP). It also supports creating thin volumes and specifying volume media affinity.

Requirements

ISE FW 2.8.0 or ISE FW 3.1.0 is required for OpenStack Block Storage support. The X-IO volume driver will not work with older ISE FW.

Supported operations

- Create, delete, attach, detach, retype, clone, and extend volumes.
- Create a volume from snapshot.
- Create, list, and delete volume snapshots.
- Manage and unmanage a volume.
- Get volume statistics.
- Create a thin provisioned volume.
- Create volumes with QoS specifications.

Configure X-IO Volume driver

To configure the use of an ISE product with OpenStack Block Storage, modify your `cinder.conf` file as follows. Be careful to use the one that matches the storage protocol in use:

Fibre Channel

```
volume_driver = cinder.volume.drivers.xio.XIOISEFCDriver
san_ip = 1.2.3.4              # the address of your ISE REST management
 interface
san_login = administrator     # your ISE management admin login
san_password = password       # your ISE management admin password
```

iSCSI

```
volume_driver = cinder.volume.drivers.xio.XIOISEISCSIDriver
san_ip = 1.2.3.4              # the address of your ISE REST management
 interface
san_login = administrator     # your ISE management admin login
san_password = password       # your ISE management admin password
iscsi_ip_address = ionet_ip   # ip address to one ISE port connected to the
 IONET
```

Optional configuration parameters

Table 2.34. Description of X-IO volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`driver_use_ssl = False`	(BoolOpt) Tell driver to use SSL for connection to backend storage if the driver supports it.
`ise_completion_retries = 30`	(IntOpt) Number on retries to get completion status after issuing a command to ISE.
`ise_connection_retries = 5`	(IntOpt) Number of retries (per port) when establishing connection to ISE management port.
`ise_raid = 1`	(IntOpt) Raid level for ISE volumes.
`ise_retry_interval = 1`	(IntOpt) Interval (secs) between retries.
`ise_storage_pool = 1`	(IntOpt) Default storage pool for volumes.

Multipath

The X-IO ISE supports a multipath configuration, but multipath must be enabled on the compute node (see *ISE Storage Blade Best Practices Guide*). For more information, see *www.openstack.org*.

Volume types

OpenStack Block Storage uses volume types to help the administrator specify attributes for volumes. These attributes are called extra-specs. The X-IO volume driver support the following extra-specs.

Table 2.35. Extra specs

Extra-specs name	Valid values	Description
Feature:Raid	1, 5	RAID level for volume.
Feature:Pool	1 - n (n being number of pools on ISE)	Pool to create volume in.
Affinity:Type	cadp, flash, hdd	Volume media affinity type.
Alloc:Type	0 (thick), 1 (thin)	Allocation type for volume. Thick or thin
QoS:minIOPS	n (value less than maxIOPS)	Minimum IOPS setting for volume.
QoS:maxIOPS	n (value bigger than minIOPS)	Maximum IOPS setting for volume.
QoS:burstIOPS	n (value bigger than minIOPS)	Burst IOPS setting for volume.

Examples

Create a volume type called xio1-flash for volumes that should reside on ssd storage:

```
$ cinder type-create xio1-flash
$ cinder type-key xio1-flash set Affinity:Type=flash
```

Create a volume type called xio1 and set QoS min and max:

```
$ cinder type-create xio1
$ cinder type-key xio1 set QoS:minIOPS=20
$ cinder type-key xio1 set QoS:maxIOPS=5000
```

Oracle ZFS Storage Appliance NFS driver

The Oracle ZFS Storage Appliance (ZFSSA) NFS driver enables the ZFSSA to be used seamlessly as a block storage resource. The driver enables you to to create volumes on a ZFS share that is NFS mounted.

Requirements

Oracle ZFS Storage Appliance Software version `2013.1.2.0` or later

Supported operations

- Create, extend, delete volumes
- Attach and detach volumes
- Create, delete snapshots

- Create a volume from a snapshot
- Copy an image to a volume
- Copy a volume to an image
- Clone a volume

Appliance configuration

Appliance configuration using the command line interface (CLI) is described below. To access the CLI, ensure SSH remote access is enabled, which is the default. You can also perform configuration using the browser user interface (BUI) or the RESTful API. Please refer to the Oracle ZFS Storage Appliance documentation for details on how to configure the Oracle ZFS Storage Appliance using the BUI, CLI and RESTful API.

1. Log in to the Oracle ZFS Storage Appliance CLI and enable the REST service. REST service needs to stay online for this driver to function.

```
zfssa:>configuration services rest enable
```

2. Create a new storage pool on the appliance if you do not want to use an existing one. This storage pool is named `'mypool'` for the sake of this documentation.

3. Create a new project and share in the storage pool (`mypool`) if you do not want to use existing ones. This driver will create a project and share by the names specified in `cinder.conf`, if the a project or share by that name doesnt already exist in the storage pool (`mypool`). The project and share are named 'NFSProject' and 'nfs_share' in the sample `cinder.conf` entries below.

4. To perform driver operations, create a role with the following authorizations:

   - `scope=svc - allow_administer=true, allow_restart=true, allow_configure=true`
   - `scope=nas - pool=pool_name, project=project_name, share=share_name, allow_clone=true, allow_createProject=true, allow_createShare=true, allow_changeSpaceProps=true, allow_changeGeneralProps=true, allow_destroy=true, allow_rollback=true, allow_takeSnap=true`

   The following examples show how to create a role with authorizations.

```
zfssa:> configuration roles
zfssa:configuration roles> role OpenStackRole
zfssa:configuration roles OpenStackRole (uncommitted)> set description=
"OpenStack NFS Cinder Driver"
zfssa:configuration roles OpenStackRole (uncommitted)> commit
zfssa:configuration roles> select OpenStackRole
zfssa:configuration roles OpenStackRole> authorizations create
zfssa:configuration roles OpenStackRole auth (uncommitted)> set scope=svc
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_administer=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_restart=true
```

```
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_configure=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> commit
```

```
zfssa:> configuration roles OpenStackRole authorizations> set scope=nas
```

The following properties need to be set when the scope of this role needs to be limited to a pool (`mypool`), a project (`NFSProject`) and a share (`nfs_share`) created in the steps above. This will prevent the user assigned to this role from being used to modify other pools, projects and shares.

```
zfssa:configuration roles OpenStackRole auth (uncommitted)> set pool=
mypool
zfssa:configuration roles OpenStackRole auth (uncommitted)> set project=
NFSProject
zfssa:configuration roles OpenStackRole auth (uncommitted)> set share=
nfs_share
```

The following properties only need to be set when a share or a project has not been created following the steps above and wish to allow the driver to create them for you.

```
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_createProject=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_createShare=true
```

```
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_clone=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_changeSpaceProps=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_destroy=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_rollback=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> set
 allow_takeSnap=true
zfssa:configuration roles OpenStackRole auth (uncommitted)> commit
```

5. Create a new user or modify an existing one and assign the new role to the user.

   The following example shows how to create a new user and assign the new role to the user.

```
zfssa:> configuration users
zfssa:configuration users> user cinder
zfssa:configuration users cinder (uncommitted)> set fullname="OpenStack
 Cinder Driver"
zfssa:configuration users cinder (uncommitted)> set initial_password=12345
zfssa:configuration users cinder (uncommitted)> commit
zfssa:configuration users> select cinder set roles=OpenStackRole
```

6. Ensure that NFS and HTTP services on the appliance are online. Note the HTTPS port number for later entry in the cinder service configuration file (`cinder.conf`). This driver uses WebDAV over HTTPS to create snapshots and clones of volumes, and therefore needs to have the HTTP service online.

   The following example illustrates enabling the services and showing their properties.

```
zfssa:> configuration services nfs
```

```
zfssa:configuration services nfs> enable
zfssa:configuration services nfs> show
Properties:
<status>= online
...
```

```
zfssa:configuration services http> enable
zfssa:configuration services http> show
Properties:
<status>= online
require_login = true
protocols = http/https
listen_port = 80
https_port = 443
```

7. Create a network interface to be used exclusively for data. An existing network interface may also be used. The following example illustrates how to make a network interface for data traffic flow only.

Note

For better performance and reliability, it is recommended to configure a separate subnet exclusively for data traffic in your cloud environment.

```
zfssa:> configuration net interfaces
zfssa:configuration net interfaces> select igbx
zfssa:configuration net interfaces igbx> set admin=false
zfssa:configuration net interfaces igbx> commit
```

8. For clustered controller systems, the following verification is required in addition to the above steps. Skip this step if a standalone system is used.

```
zfssa:> configuration cluster resources list
```

Verify that both the newly created pool and the network interface are of type "`singleton`" and are not locked to the current controller. This approach ensures that the pool and the interface used for data always belong to the active controller, regardless of the current state of the cluster. Verify that both the network interface used for management and data, and the storage pool belong to the same head.

Note

There will be a short service interruption during failback/takeover, but once the process is complete, the driver should be able to access the ZFSSA for data as well as for management.

Cinder service configuration

1. Define the following required properties in the `cinder.conf` configuration file:

```
volume_driver = cinder.volume.drivers.zfssa.zfssanfs.ZFSSANFSDriver
san_ip = myhost
san_login = username
san_password = password
zfssa_data_ip = mydata
zfssa_nfs_pool = mypool
```

Note

Management interface `san_ip` can be used instead of `zfssa_data_ip`, but it is not recommended.

2. You can also define the following additional properties in the `cinder.conf` configuration file:

```
zfssa_nfs_project = NFSProject
zfssa_nfs_share = nfs_share
zfssa_nfs_mount_options =
zfssa_nfs_share_compression = off
zfssa_nfs_share_logbias = latency
zfssa_https_port = 443
```

Note

The driver does not use the file specified in the `nfs_shares_config` option.

Driver options

The Oracle ZFS Storage Appliance NFS driver supports these options:

Table 2.36. Description of ZFS Storage Appliance NFS driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`zfssa_data_ip` = *None*	(StrOpt) Data path IP address
`zfssa_https_port` = *443*	(StrOpt) HTTPS port number
`zfssa_nfs_mount_options` =	(StrOpt) Options to be passed while mounting share over nfs
`zfssa_nfs_pool` =	(StrOpt) Storage pool name.
`zfssa_nfs_project` = *NFSProject*	(StrOpt) Project name.
`zfssa_nfs_share` = *nfs_share*	(StrOpt) Share name.
`zfssa_nfs_share_compression` = *off*	(StrOpt) Data compression.
`zfssa_nfs_share_logbias` = *latency*	(StrOpt) Synchronous write bias-latency, throughput.
`zfssa_rest_timeout` = *None*	(IntOpt) REST connection timeout. (seconds)

This driver shares additional NFS configuration options with the generic NFS driver. For a description of these, see Table 2.23, “Description of NFS storage configuration options” [94].

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://git.openstack.org/cgit/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). You can also perform 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 2.37. Description of Ceph backup driver configuration options

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 2.38. Description of IBM Tivoli Storage Manager backup driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`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 the swift back end performs a volume backup to an 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 2.39. Description of Swift backup driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`backup_swift_auth` = *per_user*	(StrOpt) Swift authentication mechanism
`backup_swift_auth_version` = *1*	(StrOpt) Swift authentication version. Specify "1" for auth 1.0, or "2" for auth 2.0

Configuration option = Default value	Description
backup_swift_block_size = *32768*	(IntOpt) The size in bytes that changes are tracked for incremental backups. backup_swift_object_size has to be multiple of backup_swift_block_size.
backup_swift_container = *volumebackups*	(StrOpt) The default Swift container to use
backup_swift_enable_progress_timer = *True*	(BoolOpt) Enable or Disable the timer to send the periodic progress notifications to Ceilometer when backing up the volume to the Swift backend storage. The default value is True to enable the timer.
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_tenant = *None*	(StrOpt) Swift tenant/account name. Required when connecting to an auth 2.0 system
backup_swift_url = *None*	(StrOpt) The URL of the Swift endpoint
backup_swift_user = *None*	(StrOpt) Swift user name
swift_catalog_info = *object-store:swift:publicURL*	(StrOpt) Info to match when looking for swift in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type> - Only used if backup_swift_url is unset

To enable the swift backup driver for 1.0 or 2.0 authentication version, specify 1 or 2 correspondingly. For example:

```
backup_swift_auth_version = 2
```

In addition, the 2.0 authentication system requires *backup_swift_tenant* setting:

```
backup_swift_tenant = <None>
```

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_auth_version = 1
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
```

NFS backup driver

The backup driver for the NFS back end backs up volumes of any type to an NFS exported backup repository.

To enable the NFS backup driver, include the following option in the [DEFAULT] section of the cinder.conf file:

```
backup_driver = cinder.backup.drivers.nfs
```

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

Table 2.40. Description of NFS backup driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
backup_container = *None*	(StrOpt) Custom container to use for backups.
backup_enable_progress_timer = *True*	(BoolOpt) Enable or Disable the timer to send the periodic progress notifications to Ceilometer when backing up the volume to the backend storage. The default value is True to enable the timer.
backup_file_size = *1999994880*	(IntOpt) The maximum size in bytes of the files used to hold backups. If the volume being backed up exceeds this size, then it will be backed up into multiple files. backup_file_size must be a multiple of backup_sha_block_size_bytes.
backup_mount_options = *None*	(StrOpt) Mount options passed to the NFS client. See NFS man page for details.
backup_mount_point_base = *$state_path/backup_mount*	(StrOpt) Base dir containing mount point for NFS share.
backup_sha_block_size_bytes = *32768*	(IntOpt) The size in bytes that changes are tracked for incremental backups. backup_file_size has to be multiple of backup_sha_block_size_bytes.
backup_share = *None*	(StrOpt) NFS share in fqdn:path, ipv4addr:path, or "[ipv6addr]:path" format.

Block Storage sample configuration files

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

cinder.conf

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.

The `cinder.conf` file contains most of the options to configure the Block Storage service.

```
                [DEFAULT]

#
# Options defined in oslo.messaging
#

# 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=local

# 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 thread 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.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 consistencygroups allowed per project (integer
# value)
#quota_consistencygroups=10

# Total amount of storage, in gigabytes, allowed for volumes
# and snapshots per project (integer value)
#quota_gigabytes=1000

# Number of volume backups allowed per project (integer value)
#quota_backups=10

# Total amount of storage, in gigabytes, allowed for backups
# per project (integer value)
#quota_backup_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

# Enables or disables use of default quota class with default
# quota. (boolean value)
#use_default_quota_class=true


#
# Options defined in cinder.service
#

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

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

# Range, in 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 on which OpenStack Volume API listens (string
# value)
#osapi_volume_listen=0.0.0.0

# Port on which OpenStack Volume API listens (integer value)
#osapi_volume_listen_port=8776

# Number of workers for OpenStack Volume API service. The
# default is equal to the number of CPUs available. (integer
# value)
#osapi_volume_workers=<None>


#
# Options defined in cinder.ssh_utils
#

# Option to enable strict host key checking.  When set to
# "True" Cinder will only connect to systems with a host key
# present in the configured "ssh_hosts_key_file".  When set to
# "False" the host key will be saved upon first connection and
# used for subsequent connections.  Default=False (boolean
# value)
#strict_ssh_host_key_policy=false

# File containing SSH host keys for the systems with which
# Cinder needs to communicate.  OPTIONAL:
# Default=$state_path/ssh_known_hosts (string value)
#ssh_hosts_key_file=$state_path/ssh_known_hosts


#
# 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

# Timeout for client connections' socket operations. If an
# incoming connection is idle for this number of seconds it
# will be closed. A value of '0' means wait forever. (integer
# value)
#client_socket_timeout=900

# 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_KEEPALIVE (True/False) for each server
# socket. (boolean value)
#tcp_keepalive=true

# Sets the value of TCP_KEEPIDLE in seconds for each server
# socket. Not supported on OS X. (integer value)
#tcp_keepidle=600

# Sets the value of TCP_KEEPINTVL in seconds for each server
# socket. Not supported on OS X. (integer value)
#tcp_keepalive_interval=<None>

# Sets the value of TCP_KEEPCNT for each server socket. Not
# supported on OS X. (integer value)
#tcp_keepalive_count=<None>

# 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 that a collection resource
# returns in a single response (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.api.views.versions
#

# Public url to use for versions endpoint. The default is
# None, which will use the request's host_url attribute to
# populate the URL base. If Cinder is operating behind a
# proxy, you will want to change this to represent the proxy's
# URL. (string value)
#public_endpoint=<None>


#
# Options defined in cinder.backup.chunkeddriver
#

# Compression algorithm (None to disable) (string value)
#backup_compression_algorithm=zlib


#
# 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=2

# The number of chunks or objects, for which one Ceilometer
# notification will be sent (integer value)
#backup_object_number_per_notification=10

# Interval, in seconds, between two progress notifications
# reporting the backup status (integer value)
#backup_timer_interval=120


#
# 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.nfs
#

# The maximum size in bytes of the files used to hold backups.
# If the volume being backed up exceeds this size, then it
# will be backed up into multiple files. (integer value)
#backup_file_size=1999994880

# The size in bytes that changes are tracked for incremental
# backups. backup_swift_object_size has to be multiple of
# backup_swift_block_size. (integer value)
#backup_sha_block_size_bytes=32768

# Enable or Disable the timer to send the periodic progress
# notifications to Ceilometer when backing up the volume to
# the backend storage. The default value is True to enable the
# timer. (boolean value)
#backup_enable_progress_timer=true

# Base dir containing mount point for NFS share. (string
# value)
#backup_mount_point_base=$state_path/backup_mount

# NFS share in fqdn:path, ipv4addr:path, or "[ipv6addr]:path"
# format. (string value)
#backup_share=<None>

# Mount options passed to the NFS client. See NFS man page for
# details. (string value)
#backup_mount_options=<None>

# Custom container to use for backups. (string value)
#backup_container=<None>


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

# The URL of the Swift endpoint (string value)
#backup_swift_url=<None>

# Info to match when looking for swift in the service catalog.
# Format is: separated values of the form:
# <service_type>:<service_name>:<endpoint_type> - Only used if
# backup_swift_url is unset (string value)
#swift_catalog_info=object-store:swift:publicURL
```

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

# Swift authentication version. Specify "1" for auth 1.0, or
# "2" for auth 2.0 (string value)
#backup_swift_auth_version=1

# Swift tenant/account name. Required when connecting to an
# auth 2.0 system (string value)
#backup_swift_tenant=<None>

# 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 size in bytes that changes are tracked for incremental
# backups. backup_swift_object_size has to be multiple of
# backup_swift_block_size. (integer value)
#backup_swift_block_size=32768

# 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

# Enable or Disable the timer to send the periodic progress
# notifications to Ceilometer when backing up the volume to
# the Swift backend storage. The default value is True to
# enable the timer. (boolean value)
#backup_swift_enable_progress_timer=true


#
# 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.cmd.volume
#

# Backend override of host value. (string value)
# Deprecated group/name - [DEFAULT]/host
#backend_host=<None>


#
# Options defined in cinder.cmd.volume_usage_audit
#

# If this option is specified then the start time specified is
# used instead of the start time of the last completed audit
# period. (string value)
#start_time=<None>

# If this option is specified then the end time specified is
# used instead of the end time of the last completed audit
# period. (string value)
#end_time=<None>

# Send the volume and snapshot create and delete notifications
# generated in the specified period. (boolean value)
#send_actions=false


#
# 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 host name 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

# Enables or disables negotiation of SSL layer compression. In
# some cases disabling compression can improve data
# throughput, such as when high network bandwidth is available
# and you use compressed image formats like qcow2. (boolean
# value)
#glance_api_ssl_compression=false

# Location of ca certificates file to use for glance client
# requests. (string value)
#glance_ca_certificates_file=<None>

# 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 that scheduler nodes listen on (string value)
#scheduler_topic=cinder-scheduler

# The topic that volume nodes listen on (string value)
#volume_topic=cinder-volume

# The topic that volume backup nodes listen on (string value)
#backup_topic=cinder-backup

# DEPRECATED: Deploy v1 of the Cinder API. (boolean value)
#enable_v1_api=true

# Deploy v2 of the Cinder API. (boolean value)
#enable_v2_api=true

# Enables or disables rate limit of 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 host name, FQDN, or IP address. (string
# value)
#host=cinder

# Availability zone of this node (string value)
#storage_availability_zone=nova

# Default availability zone for new volumes. If not set, the
# storage_availability_zone option value is used as the
# default for new volumes. (string value)
#default_availability_zone=<None>

# Default volume type to use (string value)
#default_volume_type=<None>

# Time period for which to generate volume usages. The options
# are 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 a service to be
# considered up (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

# The full class name of the volume replication API class
# (string value)
#replication_api_class=cinder.replication.api.API

# The full class name of the consistencygroup API class
# (string value)
#consistencygroup_api_class=cinder.consistencygroup.api.API

# OpenStack privileged account username. Used for requests to
# other services (such as Nova) that require an account with
# special rights. (string value)
#os_privileged_user_name=<None>

# Password associated with the OpenStack privileged account.
# (string value)
#os_privileged_user_password=<None>

# Tenant name associated with the OpenStack privileged
# account. (string value)
#os_privileged_user_tenant=<None>


#
# Options defined in cinder.compute
#

# The full class name of the compute API class to use (string
# value)
#compute_api_class=cinder.compute.nova.API


#
# Options defined in cinder.compute.nova
#

# Match this value when searching for nova in the service
# catalog. Format is: separated values of the form:
# <service_type>:<service_name>:<endpoint_type> (string value)
#nova_catalog_info=compute:Compute Service:publicURL

# Same as nova_catalog_info, but for admin endpoint. (string
# value)
#nova_catalog_admin_info=compute:Compute Service:adminURL

# Override service catalog lookup with template for nova
# endpoint e.g. http://localhost:8774/v2/%(project_id)s
# (string value)
#nova_endpoint_template=<None>

# Same as nova_endpoint_template, but for admin endpoint.
# (string value)
#nova_endpoint_admin_template=<None>

# Region name of this node (string value)
#os_region_name=<None>
```

```
# Location of ca certificates file to use for nova client
# requests. (string value)
#nova_ca_certificates_file=<None>

# Allow to perform insecure SSL requests to nova (boolean
# value)
#nova_api_insecure=false


#
# Options defined in cinder.db.api
#

# Services to be added to the available pool on create
# (boolean value)
#enable_new_services=true

# Template string to be used to generate volume names (string
# value)
#volume_name_template=volume-%s

# Template string to be used to generate snapshot names
# (string value)
#snapshot_name_template=snapshot-%s

# Template string to be used to generate backup names (string
# value)
#backup_name_template=backup-%s


#
# Options defined in cinder.db.base
#

# Driver to use for database access (string value)
#db_driver=cinder.db


#
# Options defined in cinder.image.glance
#

# Default core properties of image (list value)
#glance_core_properties=checksum,container_format,disk_format,image_name,
image_id,min_disk,min_ram,name,size

# 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.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.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.openstack.common.policy
#

# The JSON file that defines policies. (string value)
#policy_file=policy.json

# Default rule. Enforced when a requested rule is not found.
# (string value)
#policy_default_rule=default

# Directories where policy configuration files are stored.
# They can be relative to any directory in the search path
# defined by the config_dir option, or absolute paths. The
# file defined by policy_file must exist for these directories
# to be searched.  Missing or empty directories are ignored.
# (multi valued)
#policy_dirs=policy.d


#
# Options defined in cinder.openstack.common.versionutils
#

# Enables or disables fatal status of deprecations. (boolean
# value)
#fatal_deprecations=false


#
# 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.scheduler.weights.volume_number
#

# Multiplier used for weighing volume number. Negative numbers
# mean to spread vs stack. (floating point value)
#volume_number_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
#

# Cache volume availability zones in memory for the provided
# duration in seconds (integer value)
#az_cache_duration=3600

# 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

# This option is deprecated and unused. It will be removed in
# the Liberty release. (integer value)
#iser_num_targets=<None>

# 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

# The name of the 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

# This option is deprecated and unused. It will be removed in
# the Liberty release. (integer value)
#iscsi_num_targets=<None>

# 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 list of secondary IP addresses of the iSCSI daemon (list
# value)
#iscsi_secondary_ip_addresses=

# 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

# If this is set to True, attachment of volumes for image
# transfer will be aborted when multipathd is not running.
# Otherwise, it will fallback to single path. (boolean value)
#enforce_multipath_for_image_xfer=false

# Method used to wipe old volumes (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. tgtadm is default, use
# lioadm for LIO iSCSI support, scstadmin for SCST target
# support, iseradm for the ISER protocol, ietadm for iSCSI
# Enterprise Target, iscsictl for Chelsio iSCSI Target or fake
# for testing. (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

# Chiscsi (CXT) global defaults configuration file (string
# value)
#chiscsi_conf=/etc/chelsio-iscsi/chiscsi.conf

# This option is deprecated and unused. It will be removed in
# the next release. (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

# The blkio cgroup name to be used to limit bandwidth of
# volume copy (string value)
#volume_copy_blkio_cgroup_name=cinder-volume-copy

# The upper limit of bandwidth of volume copy. 0 => unlimited
# (integer value)
#volume_copy_bps_limit=0

# Sets the behavior of the iSCSI target to either perform
# write-back(on) or write-through(off). This parameter is
# valid if iscsi_helper is set to tgtadm or iseradm. (string
# value)
#iscsi_write_cache=on

# Determines the iSCSI protocol for new iSCSI volumes, created
# with tgtadm or lioadm target helpers. In order to enable
# RDMA, this parameter should be set with the value "iser".
# The supported iSCSI protocol values are "iscsi" and "iser".
# (string value)
#iscsi_protocol=iscsi

# The path to the client certificate key for verification, if
# the driver supports it. (string value)
#driver_client_cert_key=<None>
```

```
# The path to the client certificate for verification, if the
# driver supports it. (string value)
#driver_client_cert=<None>

# Tell driver to use SSL for connection to backend storage if
# the driver supports it. (boolean value)
#driver_use_ssl=false

# Float representation of the over subscription ratio when
# thin provisioning is involved. Default ratio is 20.0,
# meaning provisioned capacity can be 20 times of the total
# physical capacity. If the ratio is 10.5, it means
# provisioned capacity can be 10.5 times of the total physical
# capacity. A ratio of 1.0 means provisioned capacity cannot
# exceed the total physical capacity. A ratio lower than 1.0
# will be ignored and the default value will be used instead.
# (floating point value)
#max_over_subscription_ratio=20.0

# Certain ISCSI targets have predefined target names, SCST
# target driver uses this name. (string value)
#scst_target_iqn_name=<None>

# SCST target implementation can choose from multiple SCST
# target drivers. (string value)
#scst_target_driver=iscsi

# Option to enable/disable CHAP authentication for targets.
# (boolean value)
# Deprecated group/name - [DEFAULT]/eqlx_use_chap
#use_chap_auth=false

# CHAP user name. (string value)
# Deprecated group/name - [DEFAULT]/eqlx_chap_login
#chap_username=

# Password for specified CHAP account name. (string value)
# Deprecated group/name - [DEFAULT]/eqlx_chap_password
#chap_password=

# Namespace for driver private data values to be saved in.
# (string value)
#driver_data_namespace=<None>

# String representation for an equation that will be used to
# filter hosts. Only used when the driver filter is set to be
# used by the Cinder scheduler. (string value)
#filter_function=<None>

# String representation for an equation that will be used to
# determine the goodness of a host. Only used when using the
# goodness weigher is set to be used by the Cinder scheduler.
# (string value)
#goodness_function=<None>


#
# Options defined in cinder.volume.drivers.block_device
#
```

```
# List of all available devices (list value)
#available_devices=



#
# Options defined in cinder.volume.drivers.cloudbyte.options
#

# These values will be used for CloudByte storage's addQos API
# call. (dict value)
#cb_add_qosgroup=latency:15,iops:10,graceallowed:false,iopscontrol:true,
memlimit:0,throughput:0,tpcontrol:false,networkspeed:0

# Driver will use this API key to authenticate against the
# CloudByte storage's management interface. (string value)
#cb_apikey=None

# CloudByte storage specific account name. This maps to a
# project name in OpenStack. (string value)
#cb_account_name=None

# This corresponds to the name of Tenant Storage Machine (TSM)
# in CloudByte storage. A volume will be created in this TSM.
# (string value)
#cb_tsm_name=None

# A retry value in seconds. Will be used by the driver to
# check if volume creation was successful in CloudByte
# storage. (integer value)
#cb_confirm_volume_create_retry_interval=5

# Will confirm a successful volume creation in CloudByte
# storage by making this many number of attempts. (integer
# value)
#cb_confirm_volume_create_retries=3

# These values will be used for CloudByte storage's
# createVolume API call. (dict value)
#cb_create_volume=compression:off,deduplication:off,blocklength:512B,
sync:always,protocoltype:ISCSI,recordsize:16k



#
# Options defined in cinder.volume.drivers.datera
#

# DEPRECATED: This will be removed in the Liberty release. Use
# san_login and san_password instead. This directly sets the
# Datera API token. (string value)
#datera_api_token=<None>

# Datera API port. (string value)
#datera_api_port=7717

# Datera API version. (string value)
#datera_api_version=1

# Number of replicas to create of an inode. (string value)
#datera_num_replicas=3
```

```
#
# Options defined in cinder.volume.drivers.dell.dell_storagecenter_common
#

# Storage Center System Serial Number (integer value)
#dell_sc_ssn=64702

# Dell API port (integer value)
#dell_sc_api_port=3033

# Name of the server folder to use on the Storage Center
# (string value)
#dell_sc_server_folder=openstack

# Name of the volume folder to use on the Storage Center
# (string value)
#dell_sc_volume_folder=openstack


#
# Options defined in cinder.volume.drivers.emc.emc_vmax_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
#

# VNX authentication scope type. (string value)
#storage_vnx_authentication_type=global

# Directory path that contains the VNX security file. Make
# sure the security file is generated first. (string value)
#storage_vnx_security_file_dir=<None>

# Naviseccli Path. (string value)
#naviseccli_path=

# Storage pool name. (string value)
#storage_vnx_pool_name=<None>

# VNX secondary SP IP Address. (string value)
#san_secondary_ip=<None>

# Default timeout for CLI operations in minutes. For example,
# LUN migration is a typical long running operation, which
# depends on the LUN size and the load of the array. An upper
# bound in the specific deployment can be set to avoid
# unnecessary long wait. By default, it is 365 days long.
# (integer value)
#default_timeout=525600

# Default max number of LUNs in a storage group. By default,
# the value is 255. (integer value)
#max_luns_per_storage_group=255
```

```
# To destroy storage group when the last LUN is removed from
# it. By default, the value is False. (boolean value)
#destroy_empty_storage_group=false

# Mapping between hostname and its iSCSI initiator IP
# addresses. (string value)
#iscsi_initiators=

# Automatically register initiators. By default, the value is
# False. (boolean value)
#initiator_auto_registration=false

# Automatically deregister initiators after the related
# storage group is destroyed. By default, the value is False.
# (boolean value)
#initiator_auto_deregistration=false

# Report free_capacity_gb as 0 when the limit to maximum
# number of pool LUNs is reached. By default, the value is
# False. (boolean value)
#check_max_pool_luns_threshold=false

# Delete a LUN even if it is in Storage Groups. (boolean
# value)
#force_delete_lun_in_storagegroup=false


#
# Options defined in cinder.volume.drivers.emc.xtremio
#

# XMS cluster id in multi-cluster environment (string value)
#xtremio_cluster_name=


#
# Options defined in cinder.volume.drivers.eqlx
#

# Group name to use for creating volumes. Defaults to
# "group-0". (string value)
#eqlx_group_name=group-0

# Timeout for the Group Manager cli command execution. Default
# is 30. (integer value)
#eqlx_cli_timeout=30

# Maximum retry count for reconnection. Default is 5. (integer
# value)
#eqlx_cli_max_retries=5

# Use CHAP authentication for targets. Note that this option
# is deprecated in favour of "use_chap_auth" as specified in
# cinder/volume/driver.py and will be removed in next release.
# (boolean value)
#eqlx_use_chap=false

# Existing CHAP account name. Note that this option is
# deprecated in favour of "chap_username" as specified in
```

```
# cinder/volume/driver.py and will be removed in next release.
# (string value)
#eqlx_chap_login=admin

# Password for specified CHAP account name. Note that this
# option is deprecated in favour of "chap_password" as
# specified in cinder/volume/driver.py and will be removed in
# the next release (string value)
#eqlx_chap_password=password

# Pool in which volumes will be created. Defaults to
# "default". (string value)
#eqlx_pool=default


#
# Options defined in cinder.volume.drivers.glusterfs
#

# File with the list of available gluster shares (string
# value)
#glusterfs_shares_config=/etc/cinder/glusterfs_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)
#glusterfs_sparsed_volumes=true

# Create volumes as QCOW2 files rather than raw files.
# (boolean value)
#glusterfs_qcow2_volumes=false

# Base dir containing mount points for gluster shares. (string
# value)
#glusterfs_mount_point_base=$state_path/mnt


#
# Options defined in cinder.volume.drivers.hds.hds
#

# The configuration file for the Cinder HDS driver for HUS
# (string value)
#hds_cinder_config_file=/opt/hds/hus/cinder_hus_conf.xml


#
# Options defined in cinder.volume.drivers.hds.iscsi
#

# Configuration file for HDS iSCSI cinder plugin (string
# value)
#hds_hnas_iscsi_config_file=/opt/hds/hnas/cinder_iscsi_conf.xml


#
# Options defined in cinder.volume.drivers.hds.nfs
#

# Configuration file for HDS NFS cinder plugin (string value)
```

```
#hds_hnas_nfs_config_file=/opt/hds/hnas/cinder_nfs_conf.xml


#
# Options defined in cinder.volume.drivers.hitachi.hbsd_common
#

# Serial number of storage system (string value)
#hitachi_serial_number=<None>

# Name of an array unit (string value)
#hitachi_unit_name=<None>

# Pool ID of storage system (integer value)
#hitachi_pool_id=<None>

# Thin pool ID of storage system (integer value)
#hitachi_thin_pool_id=<None>

# Range of logical device of storage system (string value)
#hitachi_ldev_range=<None>

# Default copy method of storage system (string value)
#hitachi_default_copy_method=FULL

# Copy speed of storage system (integer value)
#hitachi_copy_speed=3

# Interval to check copy (integer value)
#hitachi_copy_check_interval=3

# Interval to check copy asynchronously (integer value)
#hitachi_async_copy_check_interval=10

# Control port names for HostGroup or iSCSI Target (string
# value)
#hitachi_target_ports=<None>

# Range of group number (string value)
#hitachi_group_range=<None>

# Request for creating HostGroup or iSCSI Target (boolean
# value)
#hitachi_group_request=false


#
# Options defined in cinder.volume.drivers.hitachi.hbsd_fc
#

# Request for FC Zone creating HostGroup (boolean value)
#hitachi_zoning_request=false


#
# Options defined in cinder.volume.drivers.hitachi.hbsd_horcm
#

# Instance numbers for HORCM (string value)
#hitachi_horcm_numbers=200,201
```

```
# Username of storage system for HORCM (string value)
#hitachi_horcm_user=<None>

# Password of storage system for HORCM (string value)
#hitachi_horcm_password=<None>

# Add to HORCM configuration (boolean value)
#hitachi_horcm_add_conf=true


#
# Options defined in cinder.volume.drivers.hitachi.hbsd_iscsi
#

# Add CHAP user (boolean value)
#hitachi_add_chap_user=false

# iSCSI authentication method (string value)
#hitachi_auth_method=<None>

# iSCSI authentication username (string value)
#hitachi_auth_user=HBSD-CHAP-user

# iSCSI authentication password (string value)
#hitachi_auth_password=HBSD-CHAP-password


#
# Options defined in cinder.volume.drivers.huawei
#

# The configuration file for the Cinder Huawei driver (string
# value)
#cinder_huawei_conf_file=/etc/cinder/cinder_huawei_conf.xml


#
# Options defined in cinder.volume.drivers.ibm.flashsystem
#

# Connection protocol should be FC. (string value)
#flashsystem_connection_protocol=FC

# Connect with multipath (FC only). (boolean value)
#flashsystem_multipath_enabled=false

# Allows vdisk to multi host mapping. (boolean value)
#flashsystem_multihostmap_enabled=true


#
# 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=system


#
# Options defined in cinder.volume.drivers.ibm.ibmnas
#

# IBMNAS platform type to be used as backend storage; valid
# values are - v7ku : for using IBM Storwize V7000 Unified,
# sonas : for using IBM Scale Out NAS, gpfs-nas : for using
# NFS based IBM GPFS deployments. (string value)
#ibmnas_platform_type=v7ku


#
# 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

# Indicate whether svc driver is compatible for NPIV setup. If
# it is compatible, it will allow no wwpns being returned on
# get_conn_fc_wwpns during initialize_connection (boolean
# value)
#storwize_svc_npiv_compatibility_mode=false

# Allow tenants to specify QOS on create (boolean value)
#storwize_svc_allow_tenant_qos=false

# If operating in stretched cluster mode, specify the name of
# the pool in which mirrored copies are stored.Example:
# "pool2" (string value)
#storwize_svc_stretched_cluster_partner=<None>


#
# 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 (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 >0, create LVs 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 (string value)
#lvm_type=default

# LVM conf file to use for the LVM driver in Cinder; this
# setting is ignored if the specified file does not exist (You
# can also specify 'None' to not use a conf file even if one
# exists). (string value)
#lvm_conf_file=/etc/cinder/lvm.conf


#
# 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>

# The name of the config.conf stanza for a Data ONTAP (7-mode)
# HA partner.  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 it is required if the storage
# protocol selected is FC. (string value)
#netapp_partner_backend_name=<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. (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. If not specified, Data ONTAP drivers
# will use 80 for HTTP and 443 for HTTPS; E-Series will use
# 8080 for HTTP and 8443 for HTTPS. (integer value)
#netapp_server_port=<None>

# 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 or FC. 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. (string value)
#netapp_storage_protocol=<None>

# The transport protocol used when communicating with the
# storage system or proxy server. (string value)
#netapp_transport_type=http


#
# Options defined in cinder.volume.drivers.nfs
#

# 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>

# The number of attempts to mount nfs shares before raising an
# error.  At least one attempt will be made to mount an nfs
# share, regardless of the value specified. (integer value)
#nfs_mount_attempts=3


#
# Options defined in cinder.volume.drivers.nimble
#

# Nimble Controller pool name (string value)
#nimble_pool_name=default

# Nimble Subnet Label (string value)
#nimble_subnet_label=*


#
# Options defined in cinder.volume.drivers.openvstorage
#

# Vpool to use for volumes - backend is defined by vpool not
# by us. (string value)
#vpool_name=


#
# Options defined in cinder.volume.drivers.prophetstor.options
#

# DPL pool uuid in which DPL volumes are stored. (string
# value)
#dpl_pool=

# DPL port number. (integer value)
#dpl_port=8357


#
# Options defined in cinder.volume.drivers.pure
#

# REST API authorization token. (string value)
#pure_api_token=<None>


#
# Options defined in cinder.volume.drivers.quobyte
#
```

```
# URL to the Quobyte volume e.g., quobyte://<DIR host>/<volume
# name> (string value)
#quobyte_volume_url=<None>

# Path to a Quobyte Client configuration file. (string value)
#quobyte_client_cfg=<None>

# Create volumes as sparse 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)
#quobyte_sparsed_volumes=true

# Create volumes as QCOW2 files rather than raw files.
# (boolean value)
#quobyte_qcow2_volumes=true

# Base dir containing the mount point for the Quobyte volume.
# (string value)
#quobyte_mount_point_base=$state_path/mnt


#
# Options defined in cinder.volume.drivers.rbd
#

# The RADOS pool where 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 (string value)
#rbd_ceph_conf=

# Flatten volumes created from snapshots to remove dependency
# from volume to snapshot (boolean value)
#rbd_flatten_volume_from_snapshot=false

# The libvirt uuid of the secret for the rbd_user volumes
# (string value)
#rbd_secret_uuid=<None>

# Directory where temporary image files are stored when the
# volume driver does not write them directly to the volume.
# Warning: this option is now deprecated, please use
# image_conversion_dir instead. (string value)
#volume_tmp_dir=<None>

# Maximum number of nested volume clones that are taken before
# a flatten occurs. Set to 0 to disable cloning. (integer
# value)
#rbd_max_clone_depth=5

# Volumes will be chunked into objects of this size (in
# megabytes). (integer value)
#rbd_store_chunk_size=4

# 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.remotefs
#

# 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=

# Allow network-attached storage systems to operate in a
# secure environment where root level access is not permitted.
# If set to False, access is as the root user and insecure. If
# set to True, access is not as root. If set to auto, a check
# is done to determine if this is a new installation: True is
# used if so, otherwise False. Default is auto. (string value)
#nas_secure_file_operations=auto

# Set more secure file permissions on network-attached storage
# volume files to restrict broad other/world access. If set to
# False, volumes are created with open permissions. If set to
# True, volumes are created with permissions for the cinder
# user and group (660). If set to auto, a check is done to
# determine if this is a new installation: True is used if so,
# otherwise False. Default is auto. (string value)
#nas_secure_file_permissions=auto

# Path to the share to use for storing Cinder volumes. For
# example:  "/srv/export1" for an NFS server export available
# at 10.0.5.10:/srv/export1 . (string value)
#nas_share_path=

# Options used to mount the storage backend file system where
# Cinder volumes are stored. (string value)
#nas_mount_options=<None>


#
# 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=

# List of the CPG(s) to use for volume creation (list value)
#hp3par_cpg=OpenStack

# The CPG to use for Snapshots for volumes. If empty the
# userCPG 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=

# Enable CHAP authentication for iSCSI connections. (boolean
# value)
#hp3par_iscsi_chap_enabled=false


#
# 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.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.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.smbfs
#

# File with the list of available smbfs shares. (string value)
#smbfs_shares_config=/etc/cinder/smbfs_shares

# Default format that will be used when creating volumes if no
```

```
# volume format is specified. (string value)
#smbfs_default_volume_format=qcow2

# Create volumes as sparsed files which take no space rather
# than regular files when using raw format, in which case
# volume creation takes lot of time. (boolean value)
#smbfs_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)
#smbfs_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)
#smbfs_oversub_ratio=1.0

# Base dir containing mount points for smbfs shares. (string
# value)
#smbfs_mount_point_base=$state_path/mnt

# Mount options passed to the smbfs client. See mount.cifs man
# page for details. (string value)
#smbfs_mount_options=noperm,file_mode=0775,dir_mode=0775


#
# 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 hostsname (previous
# default behavior).  The default is NO prefix. (string value)
#sf_account_prefix=<None>

# Account name on the SolidFire Cluster to use as owner of
# template/cache volumes (created if does not exist). (string
# value)
#sf_template_account_name=openstack-vtemplate

# Create an internal cache of copy of images when a bootable
# volume is created to eliminate fetch from glance and qemu-
# conversion on subsequent calls. (boolean value)
#sf_allow_template_caching=true

# 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.srb
```

```
#

# Comma-separated list of REST servers IP to connect to. (eg
# http://IP1/,http://IP2:81/path (string value)
#srb_base_urls=<None>


#
# Options defined in cinder.volume.drivers.violin.v6000_common
#

# IP address or hostname of mg-a (string value)
#gateway_mga=<None>

# IP address or hostname of mg-b (string value)
#gateway_mgb=<None>

# Use igroups to manage targets and initiators (boolean value)
#use_igroups=false

# Global backend request timeout, in seconds (integer value)
#request_timeout=300


#
# 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. (floating point value)
#vmware_task_poll_interval=0.5

# 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>

# Directory where virtual disks are stored during volume
# backup and restore. (string value)
#vmware_tmp_dir=/tmp


#
# 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.xio
#

# Default storage pool for volumes. (integer value)
#ise_storage_pool=1

# Raid level for ISE volumes. (integer value)
#ise_raid=1

# Number of retries (per port) when establishing connection to
# ISE management port. (integer value)
#ise_connection_retries=5

# Interval (secs) between retries. (integer value)
#ise_retry_interval=1

# Number on retries to get completion status after issuing a
# command to ISE. (integer value)
#ise_completion_retries=30


#
# Options defined in cinder.volume.drivers.zfssa.zfssanfs
#

# Data path IP address (string value)
#zfssa_data_ip=<None>

# HTTPS port number (string value)
#zfssa_https_port=443

# Options to be passed while mounting share over nfs (string
# value)
```

```
#zfssa_nfs_mount_options=

# Storage pool name. (string value)
#zfssa_nfs_pool=

# Project name. (string value)
#zfssa_nfs_project=NFSProject

# Share name. (string value)
#zfssa_nfs_share=nfs_share

# Data compression. (string value)
#zfssa_nfs_share_compression=off

# Synchronous write bias-latency, throughput. (string value)
#zfssa_nfs_share_logbias=latency

# REST connection timeout. (seconds) (integer value)
#zfssa_rest_timeout=<None>


#
# 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. The key/value pairs can be used
# by the CapabilitiesFilter to select between backends when
# requests specify volume types. For example, specifying a
# service level or the geographical location of a backend,
# then creating a volume type to allow the user to select by
# these different properties. (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>


[CISCO_FABRIC_EXAMPLE]

#
# Options defined in cinder.zonemanager.drivers.cisco.cisco_fabric_opts
#

# Management IP of fabric (string value)
#cisco_fc_fabric_address=

# Fabric user ID (string value)
#cisco_fc_fabric_user=

# Password for user (string value)
#cisco_fc_fabric_password=

# Connecting port (integer value)
#cisco_fc_fabric_port=22

# overridden zoning policy (string value)
#cisco_zoning_policy=initiator-target

# overridden zoning activation state (boolean value)
#cisco_zone_activate=true

# overridden zone name prefix (string value)
#cisco_zone_name_prefix=<None>

# VSAN of the Fabric (string value)
#cisco_zoning_vsan=<None>


[database]

#
# Options defined in oslo.db.concurrency
#

# Enable the experimental use of thread pooling for all DB API
# calls (boolean value)
# Deprecated group/name - [DEFAULT]/dbapi_use_tpool
```

```
#use_tpool=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.drivers.cisco.cisco_fc_zone_driver
#

# Southbound connector for zoning operation (string value)
#cisco_sb_connector=cinder.zonemanager.drivers.cisco.cisco_fc_zone_client_cli.
CiscoFCZoneClientCLI


#
# 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; valid values include
# "initiator-target" or "initiator" (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>


#
# Options defined in cinder.keymgr.key_mgr
#

# Authentication url for encryption service. (string value)
#encryption_auth_url=http://localhost:5000/v3

# Url for encryption service. (string value)
#encryption_api_url=http://localhost:9311/v1


[keystone_authtoken]

#
# Options defined in keystonemiddleware.auth_token
#

# 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. (integer 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

# Env key for the swift cache. (string value)
#cache=<None>

# Required if identity server requires client certificate
# (string value)
#certfile=<None>

# Required if identity 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) Number of seconds memcached server is considered
# dead before it is tried again. (integer value)
#memcache_pool_dead_retry=300

# (Optional) Maximum total number of open connections to every
# memcached server. (integer value)
#memcache_pool_maxsize=10

# (Optional) Socket timeout in seconds for communicating with
# a memcache server. (integer value)
#memcache_pool_socket_timeout=3

# (Optional) Number of seconds a connection to memcached is
# held unused in the pool before it is closed. (integer value)
#memcache_pool_unused_timeout=60

# (Optional) Number of seconds that an operation will wait to
# get a memcache client connection from the pool. (integer
# value)
#memcache_pool_conn_get_timeout=10

# (Optional) Use the advanced (eventlet safe) memcache client
```

```
# pool. The advanced pool will only work under python 2.x.
# (boolean value)
#memcache_use_advanced_pool=false

# (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
# identity 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
#

# 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 to verify 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


[oslo_messaging_qpid]

#
# 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)
#qpid_topology_version=1


[oslo_messaging_rabbit]

#
# 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

# SSL version to use (valid only if SSL enabled). Valid values
# are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may
# be available on some distributions. (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=AMQPLAIN

# 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

# Number of seconds after which the Rabbit broker is
# considered down if heartbeat's keep-alive fails (0 disables
# the heartbeat, >0 enables it. Enabling heartbeats requires
# kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL (integer value)
#heartbeat_timeout_threshold=0

# How often times during the heartbeat_timeout_threshold we
# check the heartbeat. (integer value)
#heartbeat_rate=2

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


[DEFAULT]


[keystone_authtoken]

#
# From keystonemiddleware.auth_token
#

# 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.
# (integer 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

# Env key for the swift cache. (string value)
#cache = <None>

# Required if identity server requires client certificate (string
# value)
#certfile = <None>

# Required if identity 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) Number of seconds memcached server is considered dead
# before it is tried again. (integer value)
#memcache_pool_dead_retry = 300
```

```
# (Optional) Maximum total number of open connections to every
# memcached server. (integer value)
#memcache_pool_maxsize = 10

# (Optional) Socket timeout in seconds for communicating with a
# memcache server. (integer value)
#memcache_pool_socket_timeout = 3

# (Optional) Number of seconds a connection to memcached is held
# unused in the pool before it is closed. (integer value)
#memcache_pool_unused_timeout = 60

# (Optional) Number of seconds that an operation will wait to get a
# memcache client connection from the pool. (integer value)
#memcache_pool_conn_get_timeout = 10

# (Optional) Use the advanced (eventlet safe) memcache client pool.
# The advanced pool will only work under python 2.x. (boolean value)
#memcache_use_advanced_pool = false

# (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 identity 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

# 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 admin Identity API endpoint. This should specify the
# unversioned root endpoint e.g. https://localhost:35357/ (string
# value)
#identity_uri = <None>

# 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>

# Service username. (string value)
#admin_user = <None>

# Service user password. (string value)
#admin_password = <None>

# Service tenant name. (string value)
#admin_tenant_name = admin
```

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 osprofiler noauth apiv1
keystone = request_id faultwrap sizelimit osprofiler authtoken keystonecontext
 apiv1
keystone_nolimit = request_id faultwrap sizelimit osprofiler authtoken
 keystonecontext apiv1

[composite:openstack_volume_api_v2]
use = call:cinder.api.middleware.auth:pipeline_factory
noauth = request_id faultwrap sizelimit osprofiler noauth apiv2
keystone = request_id faultwrap sizelimit osprofiler authtoken keystonecontext
 apiv2
```

```
keystone_nolimit = request_id faultwrap sizelimit osprofiler authtoken
 keystonecontext apiv2

[filter:request_id]
paste.filter_factory = oslo_middleware.request_id:RequestId.factory

[filter:faultwrap]
paste.filter_factory = cinder.api.middleware.fault:FaultWrapper.factory

[filter:osprofiler]
paste.filter_factory = osprofiler.web:WsgiMiddleware.factory
hmac_keys = SECRET_KEY
enabled = yes

[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 = keystonemiddleware.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 or project_id:%(project_id)s",
    "default": "rule:admin_or_owner",

    "admin_api": "is_admin:True",

    "volume:create": "",
```

```
    "volume:delete": "",
    "volume:get": "",
    "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_access": "",
    "volume_extension:volume_type_access:addProjectAccess": "rule:admin_api",
    "volume_extension:volume_type_access:removeProjectAccess":
"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:backup_admin_actions:reset_status": "rule:admin_api",
    "volume_extension:volume_admin_actions:force_delete": "rule:admin_api",
    "volume_extension:volume_admin_actions:force_detach": "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_extension:volume_manage": "rule:admin_api",
    "volume_extension:volume_unmanage": "rule:admin_api",

    "volume:services": "rule:admin_api",

    "volume:create_transfer": "",
    "volume:accept_transfer": "",
    "volume:delete_transfer": "",
    "volume:get_all_transfers": "",

    "volume_extension:replication:promote": "rule:admin_api",
    "volume_extension:replication:reenable": "rule:admin_api",

    "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": "",

    "consistencygroup:create" : "group:nobody",
    "consistencygroup:delete": "group:nobody",
    "consistencygroup:update": "group:nobody",
    "consistencygroup:get": "group:nobody",
    "consistencygroup:get_all": "group:nobody",

    "consistencygroup:create_cgsnapshot" : "group:nobody",
    "consistencygroup:delete_cgsnapshot": "group:nobody",
    "consistencygroup:get_cgsnapshot": "group:nobody",
    "consistencygroup:get_all_cgsnapshots": "group:nobody",

    "scheduler_extension:scheduler_stats:get_pools" : "rule:admin_api"
}
```

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
# explicitely 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,/usr/local/bin,/usr/local/sbin

# 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 2.41. 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)
`api.log`	`openstack-cinder-api`	`cinder-api`
`cinder-manage.log`	`cinder-manage`	`cinder-manage`
`scheduler.log`	`openstack-cinder-scheduler`	`cinder-scheduler`
`volume.log`	`openstack-cinder-volume`	`cinder-volume`

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 2.42. Description of zoning configuration options

Configuration option = Default value	Description
[DEFAULT]	
`zoning_mode` = *none*	(StrOpt) FC Zoning mode configured
[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_san_lookup_service` = *`cinder.zonemanager.drivers.brocade.brcd_fc_san_lookup_service.BrcdFCSanLookupService`*	(StrOpt) FC SAN Lookup Service
`zone_driver` = *`cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver.BrcdFCZoneDriver`*	(StrOpt) FC Zone Driver responsible for zone management
`zoning_policy` = *`initiator-target`*	(StrOpt) Zoning policy configured by user; valid values include "initiator-target" or "initiator"

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 through SSH. Configure Brocade Zone Driver and lookup service by specifying the following parameters:

Table 2.43. Description of zoning manager configuration options

Configuration option = Default value	Description
[fc-zone-manager]	
`brcd_sb_connector` = *`cinder.zonemanager.drivers.brocade.brcd_fc_zone_client_cli.BrcdFCZoneClientCLI`*	(StrOpt) Southbound connector for zoning operation

Configure SAN fabric parameters in the form of fabric groups as described in the example below:

Table 2.44. Description of zoning fabrics configuration options

Configuration option = Default value	Description
[BRCD_FABRIC_EXAMPLE]	
`fc_fabric_address` =	(StrOpt) Management IP of fabric
`fc_fabric_password` =	(StrOpt) Password for user
`fc_fabric_port` = *`22`*	(IntOpt) Connecting port
`fc_fabric_user` =	(StrOpt) Fabric user ID
`principal_switch_wwn` = *`None`*	(StrOpt) Principal switch WWN of the fabric
`zone_activate` = *`True`*	(BoolOpt) overridden zoning activation state
`zone_name_prefix` = *`None`*	(StrOpt) overridden zone name prefix
`zoning_policy` = *`initiator-target`*	(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.

Cisco Fibre Channel Zone Driver

Cisco Fibre Channel Zone Driver automates the zoning operations through SSH. Configure Cisco Zone Driver, Cisco Southbound connector, FC SAN lookup service and Fabric name.

Set the following options in the `cinder.conf` configuration file.

```
[fc-zone-manager]
zone_driver = cinder.zonemanager.drivers.cisco.cisco_fc_zone_driver.
CiscoFCZoneDriver
fc_san_lookup_service = cinder.zonemanager.drivers.cisco.
cisco_fc_san_lookup_service.CiscoFCSanLookupService
fc_fabric_names = CISCO_FABRIC_EXAMPLE
cisco_sb_connector = cinder.zonemanager.drivers.cisco.
cisco_fc_zone_client_cli.CiscoFCZoneClientCLI
```

Table 2.45. Description of cisco zoning manager configuration options

Configuration option = Default value	Description
[fc-zone-manager]	
`cisco_sb_connector` = *cinder.zonemanager.drivers.cisco.cisco_fc_zone_client_cli.CiscoFCZoneClientCLI*	(StrOpt) Southbound connector for zoning operation

Configure SAN fabric parameters in the form of fabric groups as described in the example below:

Table 2.46. Description of cisco zoning fabrics configuration options

Configuration option = Default value	Description
[CISCO_FABRIC_EXAMPLE]	
`cisco_fc_fabric_address` =	(StrOpt) Management IP of fabric
`cisco_fc_fabric_password` =	(StrOpt) Password for user
`cisco_fc_fabric_port` = *22*	(IntOpt) Connecting port
`cisco_fc_fabric_user` =	(StrOpt) Fabric user ID
`cisco_zone_activate` = *True*	(BoolOpt) overridden zoning activation state
`cisco_zone_name_prefix` = *None*	(StrOpt) overridden zone name prefix
`cisco_zoning_policy` = *initiator-target*	(StrOpt) overridden zoning policy
`cisco_zoning_vsan` = *None*	(StrOpt) VSAN of the Fabric

Note

Define a fabric group for each fabric using the fabric names used in `fc_fabric_names` configuration option as group name.

The Cisco Fibre Channel Zone Driver supports basic and enhanced zoning modes.The zoning VSAN must exist with an active zone set name which is same as the `fc_fabric_names` option.

System requirements

Cisco MDS 9000 Family Switches.

Cisco MDS NX-OS Release 6.2(9) or later.

For information about how to manage Cisco Fibre Channel switches, see the Cisco MDS 9000 user documentation.

Volume encryption with static key

This is an implementation of a key manager that reads its key from the project's configuration options.

This key manager implementation provides limited security, assuming that the key remains secret. Volume encryption provides protection against a lost or stolen disk, assuming that the configuration file that contains the key is not stored on the disk. Encryption also protects the confidentiality of data as it is transmitted via iSCSI from the compute host to the storage host as long as an attacker who intercepts the data does not know the secret key.

Because this implementation uses a single, fixed key, it does not provide protection if that key is compromised. In particular, different volumes encrypted with a key provided by this key manager actually share the same encryption key so *any* volume can be decrypted once the fixed key is known.

Updates are in the pipeline which will provide true key manager support via the key management service. This will provide much better security once complete.

Initial configuration

Configuration changes need to be made to any nodes running the `cinder-volume` or `nova-compute` services.

Update `cinder-volume` servers:

1. Edit the `/etc/cinder/cinder.conf` file and add or update the value of the option `fixed_key` in the `[keymgr]` section:

```
[keymgr]
# Fixed key returned by key manager, specified in hex (string
# value)
fixed_key =
 0000000000000000000000000000000000000000000000000000000000000000
```

2. Restart `cinder-volume`.

Update `nova-compute` servers:

1. Edit the `/etc/nova/nova.conf` file and add or update the value of the option `fixed_key` in the `[keymgr]` section (add a keymgr section as shown if needed):

```
[keymgr]
# Fixed key returned by key manager, specified in hex (string
# value)
fixed_key =
 0000000000000000000000000000000000000000000000000000000000000000
```

2. Restart `nova-compute`.

Create encrypted volume type

Block Storage volume type assignment provides scheduling to a specific back-end, and can be used to specify actionable information for a back-end storage device.

This example creates a volume type called LUKS and provides configuration information for the storage system to encrypt or decrypt the volume.

1. Source your admin credentials:

```
$ source admin-openrc.sh
```

2. Create the volume type:

```
$ cinder type-create LUKS
+--------------------------------------+-------+
|                  ID                  |  Name |
+--------------------------------------+-------+
| e64b35a4-a849-4c53-9cc7-2345d3c8fbde | LUKS  |
+--------------------------------------+-------+
```

3. Mark the volume type as encrypted and provide the necessary details. Use `--control_location` to specify where encryption is performed: `front-end` (default) or `back-end`.

```
$ cinder encryption-type-create --cipher aes-xts-plain64 --key_size 512 \
  --control_location front-end LUKS nova.volume.encryptors.luks.
LuksEncryptor
+--------------------------------------
+-----------------------------------------+-----------------+----------
+------------------+
|            Volume Type ID            |                  Provider
         |      Cipher     | Key Size | Control Location |
+--------------------------------------
+-----------------------------------------+-----------------+----------
+------------------+
| e64b35a4-a849-4c53-9cc7-2345d3c8fbde | nova.volume.encryptors.luks.
LuksEncryptor | aes-xts-plain64 |   512    |    front-end     |
+--------------------------------------
+-----------------------------------------+-----------------+----------
+------------------+
```

Support for creating the volume type in the OpenStack dashboard (horizon) exists today, however support for tagging the type as encrypted and providing the additional information needed is still in review.

Create an encrypted volume

Use the OpenStack dashboard (horizon), or the **cinder** command to create volumes just as you normally would. For an encrypted volume use the LUKS tag, for unencrypted leave the LUKS tag off.

1. Source your admin credentials:

```
$ source admin-openrc.sh
```

2. Create an unencrypted 1 GB test volume:

```
$ cinder create --display-name 'unencrypted volume' 1
+--------------------------------+----------------------------------------+
|            Property            |                 Value                  |
+--------------------------------+----------------------------------------+
|          attachments           |                   []                   |
```

```
|       availability_zone        |                  nova                  |
|            bootable            |                 false                  |
|           created_at           |      2014-08-10T01:24:03.000000        |
|          description           |                  None                  |
|           encrypted            |                 False                  |
|               id               | 081700fd-2357-44ff-860d-2cd78ad9c568 |
|            metadata            |                   {}                   |
|              name              |          unencrypted volume            |
|     os-vol-host-attr:host      |              controller                |
| os-vol-mig-status-attr:migstat |                  None                  |
| os-vol-mig-status-attr:name_id |                  None                  |
|  os-vol-tenant-attr:tenant_id  |   08fdea76c760475f82087a45dbe94918     |
|              size              |                   1                    |
|          snapshot_id           |                  None                  |
|          source_volid          |                  None                  |
|             status             |                creating                |
|            user_id             |   7cbc6b58b372439e8f70e2a9103f1332     |
|          volume_type           |                  None                  |
+--------------------------------+----------------------------------------+
```

3. Create an encrypted 1 GB test volume:

```
$ cinder create --display-name 'encrypted volume' --volume-type LUKS 1
+--------------------------------+----------------------------------------+
|            Property            |                 Value                  |
+--------------------------------+----------------------------------------+
|          attachments           |                   []                   |
|       availability_zone        |                  nova                  |
|            bootable            |                 false                  |
|           created_at           |      2014-08-10T01:24:24.000000        |
|          description           |                  None                  |
|           encrypted            |                  True                  |
|               id               | 86060306-6f43-4c92-9ab8-ddcd83acd973 |
|            metadata            |                   {}                   |
|              name              |           encrypted volume             |
|     os-vol-host-attr:host      |              controller                |
| os-vol-mig-status-attr:migstat |                  None                  |
| os-vol-mig-status-attr:name_id |                  None                  |
|  os-vol-tenant-attr:tenant_id  |   08fdea76c760475f82087a45dbe94918     |
|              size              |                   1                    |
|          snapshot_id           |                  None                  |
|          source_volid          |                  None                  |
|             status             |                creating                |
|            user_id             |   7cbc6b58b372439e8f70e2a9103f1332     |
|          volume_type           |                  LUKS                  |
+--------------------------------+----------------------------------------+
```

Notice the encrypted parameter; it will show True/False. The option `volume_type` is also shown for easy review.

Testing volume encryption

This is a simple test scenario to help validate your encryption. It assumes an LVM based Block Storage server.

Perform these steps after completing the volume encryption setup and creating the volume-type for LUKS as described in the preceding sections.

1. Create a VM:

```
$ nova boot --flavor m1.tiny --image cirros-0.3.1-x86_64-disk vm-test
```

2. Create two volumes, one encrypted and one not encrypted then attach them to your VM:

```
$ cinder create --display-name 'unencrypted volume' 1
$ cinder create --display-name 'encrypted volume' --volume-type LUKS 1
$ cinder list
+--------------------------------------+-----------+--------------------
+------+-------------+----------+-------------+
|                  ID                  |   Status  |        Name        |
 Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+--------------------
+------+-------------+----------+-------------+
| 64b48a79-5686-4542-9b52-d649b51c10a2 | available | unencrypted volume |
 1   |     None    |  false   |             |
| db50b71c-bf97-47cb-a5cf-b4b43a0edab6 | available |  encrypted volume  |
 1   |     LUKS    |  false   |             |
+--------------------------------------+-----------+--------------------
+------+-------------+----------+-------------+
$ nova volume-attach vm-test 64b48a79-5686-4542-9b52-d649b51c10a2 /dev/vdb
$ nova volume-attach vm-test db50b71c-bf97-47cb-a5cf-b4b43a0edab6 /dev/vdc
```

3. On the VM, send some text to the newly attached volumes and synchronize them:

```
# echo "Hello, world (unencrypted /dev/vdb)" >> /dev/vdb
# echo "Hello, world (encrypted /dev/vdc)" >> /dev/vdc
# sync && sleep 2
# sync && sleep 2
```

4. On the system hosting cinder volume services, synchronize to flush the I/O cache then test to see if your strings can be found:

```
# sync && sleep 2
# sync && sleep 2
# strings /dev/stack-volumes/volume-* | grep "Hello"
Hello, world (unencrypted /dev/vdb)
```

In the above example you see that the search returns the string written to the unencrypted volume, but not the encrypted one.

Additional options

These options can also be set in the `cinder.conf` file.

Table 2.47. Description of API configuration options

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) Enables or disables rate limit of the API.
`az_cache_duration` = *`3600`*	(IntOpt) Cache volume availability zones in memory for the provided duration in seconds
`backend_host` = *`None`*	(StrOpt) Backend override of host value.

Configuration option = Default value	Description
default_timeout = *525600*	(IntOpt) Default timeout for CLI operations in minutes. For example, LUN migration is a typical long running operation, which depends on the LUN size and the load of the array. An upper bound in the specific deployment can be set to avoid unnecessary long wait. By default, it is 365 days long.
enable_v1_api = *True*	(BoolOpt) DEPRECATED: 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. The key/value pairs can be used by the CapabilitiesFilter to select between backends when requests specify volume types. For example, specifying a service level or the geographical location of a backend, then creating a volume type to allow the user to select by these different properties.
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 that a collection resource returns in a single response
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 on which OpenStack Volume API listens
osapi_volume_listen_port = *8776*	(IntOpt) Port on which OpenStack Volume API listens
osapi_volume_workers = *None*	(IntOpt) Number of workers for OpenStack Volume API service. The default is equal to the number of CPUs available.
public_endpoint = *None*	(StrOpt) Public url to use for versions endpoint. The default is None, which will use the request's host_url attribute to populate the URL base. If Cinder is operating behind a proxy, you will want to change this to represent the proxy's URL.
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_number_multiplier = *-1.0*	(FloatOpt) Multiplier used for weighing volume number. Negative numbers mean to spread vs stack.
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 2.48. Description of AMQP configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
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.
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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 2.49. Description of authorization configuration options

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = *noauth*	(StrOpt) The strategy to use for auth. Supports noauth, keystone, and deprecated.

Table 2.50. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.

Configuration option = Default value	Description
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.

Configuration option = Default value	Description
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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 2.51. Description of backups configuration options

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 = *2*	(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_object_number_per_notification = *10*	(IntOpt) The number of chunks or objects, for which one Ceilometer notification will be sent
backup_timer_interval = *120*	(IntOpt) Interval, in seconds, between two progress notifications reporting the backup status
backup_topic = *cinder-backup*	(StrOpt) The topic that 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 2.52. Description of block device configuration options

Configuration option = Default value	Description
[DEFAULT]	
available_devices =	(ListOpt) List of all available devices

Table 2.53. Description of CA and SSL configuration options

Configuration option = Default value	Description
[DEFAULT]	
ssl_ca_file = *None*	(StrOpt) CA certificate file to use to verify connecting clients

Configuration option = Default value	Description
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 2.54. Description of CloudByte volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
cb_account_name = *None*	(StrOpt) CloudByte storage specific account name. This maps to a project name in OpenStack.
cb_add_qosgroup = *{'latency': '15', 'iops': '10', 'graceallowed': 'false', 'iopscontrol': 'true', 'memlimit': '0', 'throughput': '0', 'tpcontrol': 'false', 'networkspeed': '0'}*	(DictOpt) These values will be used for CloudByte storage's addQos API call.
cb_apikey = *None*	(StrOpt) Driver will use this API key to authenticate against the CloudByte storage's management interface.
cb_confirm_volume_create_retries = *3*	(IntOpt) Will confirm a successful volume creation in CloudByte storage by making this many number of attempts.
cb_confirm_volume_create_retry_interval = *5*	(IntOpt) A retry value in seconds. Will be used by the driver to check if volume creation was successful in CloudByte storage.
cb_create_volume = *{'compression': 'off', 'deduplication': 'off', 'blocklength': '512B', 'sync': 'always', 'protocoltype': 'ISCSI', 'recordsize': '16k'}*	(DictOpt) These values will be used for CloudByte storage's createVolume API call.
cb_tsm_name = *None*	(StrOpt) This corresponds to the name of Tenant Storage Machine (TSM) in CloudByte storage. A volume will be created in this TSM.

Table 2.55. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
chap_password =	(StrOpt) Password for specified CHAP account name.
chap_username =	(StrOpt) CHAP user name.
chiscsi_conf = */etc/chelsio-iscsi/chiscsi.conf*	(StrOpt) Chiscsi (CXT) global defaults configuration file
client_socket_timeout = *900*	(IntOpt) Timeout for client connections' socket operations. If an incoming connection is idle for this number of seconds it will be closed. A value of '0' means wait forever.
compute_api_class = *cinder.compute.nova.API*	(StrOpt) The full class name of the compute API class to use
consistencygroup_api_class = *cinder.consistencygroup.api.API*	(StrOpt) The full class name of the consistencygroup API class
default_availability_zone = *None*	(StrOpt) Default availability zone for new volumes. If not set, the storage_availability_zone option value is used as the default for new volumes.
default_volume_type = *None*	(StrOpt) Default volume type to use
driver_data_namespace = *None*	(StrOpt) Namespace for driver private data values to be saved in.
enable_new_services = *True*	(BoolOpt) Services to be added to the available pool on create

Configuration option = Default value	Description
end_time = *None*	(StrOpt) If this option is specified then the end time specified is used instead of the end time of the last completed audit period.
enforce_multipath_for_image_xfer = *False*	(BoolOpt) If this is set to True, attachment of volumes for image transfer will be aborted when multipathd is not running. Otherwise, it will fallback to single path.
host = *localhost*	(StrOpt) Name of this node. This can be an opaque identifier. It is not necessarily a host name, FQDN, or IP address.
iet_conf = */etc/iet/ietd.conf*	(StrOpt) IET configuration file
iscsi_secondary_ip_addresses =	(ListOpt) The list of secondary IP addresses of the iSCSI daemon
lio_initiator_iqns =	(StrOpt) This option is deprecated and unused. It will be removed in the next release.
max_over_subscription_ratio = *20.0*	(FloatOpt) Float representation of the over subscription ratio when thin provisioning is involved. Default ratio is 20.0, meaning provisioned capacity can be 20 times of the total physical capacity. If the ratio is 10.5, it means provisioned capacity can be 10.5 times of the total physical capacity. A ratio of 1.0 means provisioned capacity cannot exceed the total physical capacity. A ratio lower than 1.0 will be ignored and the default value will be used instead.
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
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
os_privileged_user_name = *None*	(StrOpt) OpenStack privileged account username. Used for requests to other services (such as Nova) that require an account with special rights.
os_privileged_user_password = *None*	(StrOpt) Password associated with the OpenStack privileged account.
os_privileged_user_tenant = *None*	(StrOpt) Tenant name associated with the OpenStack privileged account.
periodic_fuzzy_delay = *60*	(IntOpt) Range, in seconds, to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = *60*	(IntOpt) Interval, in seconds, between running periodic tasks
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = *['policy.d']*	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = *policy.json*	(StrOpt) The JSON file that defines policies.
replication_api_class = *cinder.replication.api.API*	(StrOpt) The full class name of the volume replication API class
report_interval = *10*	(IntOpt) Interval, in seconds, between nodes reporting state to datastore

Configuration option = Default value	Description
request_timeout = *300*	(IntOpt) Global backend request timeout, in seconds
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?
send_actions = *False*	(BoolOpt) Send the volume and snapshot create and delete notifications generated in the specified period.
service_down_time = *60*	(IntOpt) Maximum time since last check-in for a service to be considered up
sqlite_clean_db = *clean.sqlite*	(StrOpt) File name of clean sqlite db
ssh_hosts_key_file = *$state_path/ssh_known_hosts*	(StrOpt) File containing SSH host keys for the systems with which Cinder needs to communicate. OPTIONAL: Default=$state_path/ssh_known_hosts
start_time = *None*	(StrOpt) If this option is specified then the start time specified is used instead of the start time of the last completed audit period.
state_path = */var/lib/cinder*	(StrOpt) Top-level directory for maintaining cinder's state
storage_availability_zone = *nova*	(StrOpt) Availability zone of this node
strict_ssh_host_key_policy = *False*	(BoolOpt) Option to enable strict host key checking. When set to "True" Cinder will only connect to systems with a host key present in the configured "ssh_hosts_key_file". When set to "False" the host key will be saved upon first connection and used for subsequent connections. Default=False
tcp_keepalive = *True*	(BoolOpt) Sets the value of TCP_KEEPALIVE (True/False) for each server socket.
tcp_keepalive_count = *None*	(IntOpt) Sets the value of TCP_KEEPCNT for each server socket. Not supported on OS X.
tcp_keepalive_interval = *None*	(IntOpt) Sets the value of TCP_KEEPINTVL in seconds for each server socket. Not supported on OS X.
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_chap_auth = *False*	(BoolOpt) Option to enable/disable CHAP authentication for targets.
use_forwarded_for = *False*	(BoolOpt) Treat X-Forwarded-For as the canonical remote address. Only enable this if you have a sanitizing proxy.
wsgi_keep_alive = *True*	(BoolOpt) If False, closes the client socket connection explicitly. Setting it to True to maintain backward compatibility. Recommended setting is set it to False.
[keystone_authtoken]	
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.

Table 2.56. Description of Compute configuration options

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.

Configuration option = Default value	Description
nova_catalog_admin_info = *compute:Compute Service:adminURL*	(StrOpt) Same as nova_catalog_info, but for admin endpoint.
nova_catalog_info = *compute:Compute Service:publicURL*	(StrOpt) Match this value when searching 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 2.57. Description of database configuration options

Configuration option = Default value	Description
[DEFAULT]	
db_driver = *cinder.db*	(StrOpt) Driver to use for database access
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify 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.

Configuration option = Default value	Description
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.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.
use_tpool = *False*	(BoolOpt) Enable the experimental use of thread pooling for all DB API calls

Table 2.58. Description of logging configuration options

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.

Table 2.59. Description of EMC configuration options

Configuration option = Default value	Description
[DEFAULT]	
check_max_pool_luns_threshold = *False*	(BoolOpt) Report free_capacity_gb as 0 when the limit to maximum number of pool LUNs is reached. By default, the value is False.
cinder_emc_config_file = */etc/cinder/cinder_emc_config.xml*	(StrOpt) use this file for cinder emc plugin config data
destroy_empty_storage_group = *False*	(BoolOpt) To destroy storage group when the last LUN is removed from it. By default, the value is False.
force_delete_lun_in_storagegroup = *False*	(BoolOpt) Delete a LUN even if it is in Storage Groups.
initiator_auto_deregistration = *False*	(BoolOpt) Automatically deregister initiators after the related storage group is destroyed. By default, the value is False.
initiator_auto_registration = *False*	(BoolOpt) Automatically register initiators. By default, the value is False.
iscsi_initiators =	(StrOpt) Mapping between hostname and its iSCSI initiator IP addresses.
max_luns_per_storage_group = *255*	(IntOpt) Default max number of LUNs in a storage group. By default, the value is 255.
naviseccli_path =	(StrOpt) Naviseccli Path.
storage_vnx_authentication_type = *global*	(StrOpt) VNX authentication scope type.
storage_vnx_pool_name = *None*	(StrOpt) Storage pool name.
storage_vnx_security_file_dir = *None*	(StrOpt) Directory path that contains the VNX security file. Make sure the security file is generated first.
xtremio_cluster_name =	(StrOpt) XMS cluster id in multi-cluster environment

Table 2.60. Description of IBM FlashSystem volume river configuration options

Configuration option = Default value	Description
[DEFAULT]	
`flashsystem_connection_protocol = FC`	(StrOpt) Connection protocol should be FC.
`flashsystem_multihostmap_enabled = True`	(BoolOpt) Allows vdisk to multi host mapping.
`flashsystem_multipath_enabled = False`	(BoolOpt) Connect with multipath (FC only).

Table 2.61. Description of HP 3PAR Fibre Channel and iSCSI drivers configuration options

Configuration option = Default value	Description
[DEFAULT]	
`hp3par_api_url =`	(StrOpt) 3PAR WSAPI Server Url like https://<3par ip>:8080/api/v1
`hp3par_cpg = OpenStack`	(ListOpt) List of the CPG(s) to use for volume creation
`hp3par_cpg_snap =`	(StrOpt) The CPG to use for Snapshots for volumes. If empty the userCPG will be used.
`hp3par_debug = False`	(BoolOpt) Enable HTTP debugging to 3PAR
`hp3par_iscsi_chap_enabled = False`	(BoolOpt) Enable CHAP authentication for iSCSI connections.
`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 2.62. Description of HP LeftHand/StoreVirtual driver configuration options

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 2.63. Description of Huawei storage driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
`cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml`	(StrOpt) The configuration file for the Cinder Huawei driver

Table 2.64. Description of IBM NAS volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
ibmnas_platform_type = *v7ku*	(StrOpt) IBMNAS platform type to be used as backend storage; valid values are - v7ku : for using IBM Storwize V7000 Unified, sonas : for using IBM Scale Out NAS, gpfs-nas : for using NFS based IBM GPFS deployments.

Table 2.65. Description of images configuration options

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) Enables or disables negotiation of SSL layer compression. In some cases disabling compression can improve data throughput, such as when high network bandwidth is available and you use compressed image formats like qcow2.
glance_api_version = *1*	(IntOpt) Version of the glance API to use
glance_ca_certificates_file = *None*	(StrOpt) Location of ca certificates file to use for glance client requests.
glance_core_properties = *checksum, container_format, disk_format, image_name, image_id, min_disk, min_ram, name, size*	(ListOpt) Default core properties of image
glance_host = *$my_ip*	(StrOpt) Default glance host name or IP
glance_num_retries = *0*	(IntOpt) Number retries when downloading an image from glance
glance_port = *9292*	(IntOpt) Default glance port
glance_request_timeout = *None*	(IntOpt) http/https timeout value for glance operations. If no value (None) is supplied here, the glanceclient default value is used.
image_conversion_dir = *$state_path/conversion*	(StrOpt) Directory used for temporary storage during image conversion
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 2.66. Description of key manager configuration options

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
encryption_api_url = *http://localhost:9311/v1*	(StrOpt) Url for encryption service.
encryption_auth_url = *http://localhost:5000/v3*	(StrOpt) Authentication url for encryption service.
fixed_key = *None*	(StrOpt) Fixed key returned by key manager, specified in hex

Table 2.67. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
fatal_exception_format_errors = *False*	(BoolOpt) Make exception message format errors fatal.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.
publish_errors = *False*	(BoolOpt) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
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 will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 2.68. Description of NAS configuration options

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_mount_options = *None*	(StrOpt) Options used to mount the storage backend file system where Cinder volumes are stored.
nas_password =	(StrOpt) Password to connect to NAS system.
nas_private_key =	(StrOpt) Filename of private key to use for SSH authentication.
nas_secure_file_operations = *auto*	(StrOpt) Allow network-attached storage systems to operate in a secure environment where root level access is not permitted. If set to False, access is as the root user and insecure. If set to True, access is not as root. If set to auto, a check is done to determine if this is a new installation: True is used if so, otherwise False. Default is auto.
nas_secure_file_permissions = *auto*	(StrOpt) Set more secure file permissions on network-attached storage volume files to restrict broad other/world access. If set to False, volumes are created with open permissions. If set to True, volumes are created with permissions for the cinder user and group (660). If set to auto, a check is done to determine if this is a new installation: True is used if so, otherwise False. Default is auto.

Configuration option = Default value	Description
nas_share_path =	(StrOpt) Path to the share to use for storing Cinder volumes. For example: "/srv/export1" for an NFS server export available at 10.0.5.10:/srv/export1 .
nas_ssh_port = *22*	(IntOpt) SSH port to use to connect to NAS system.

Table 2.69. Description of Open vStorage driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
vpool_name =	(StrOpt) Vpool to use for volumes - backend is defined by vpool not by us.

Table 2.70. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
max_request_body_size = *114688*	(IntOpt) The maximum body size for each request, in bytes.

Table 2.71. Description of profiler configuration options

Configuration option = Default value	Description
[profiler]	
profiler_enabled = *False*	(BoolOpt) If False fully disable profiling feature.
trace_sqlalchemy = *False*	(BoolOpt) If False doesn't trace SQL requests.

Table 2.72. Description of Pure Storage driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
pure_api_token = *None*	(StrOpt) REST API authorization token.

Table 2.73. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
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_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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

Configuration option = Default value	Description
	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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 2.74. Description of quota configuration options

Configuration option = Default value	Description
[DEFAULT]	
max_age = *0*	(IntOpt) Number of seconds between subsequent usage refreshes
quota_backup_gigabytes = *1000*	(IntOpt) Total amount of storage, in gigabytes, allowed for backups per project
quota_backups = *10*	(IntOpt) Number of volume backups allowed per project
quota_consistencygroups = *10*	(IntOpt) Number of consistencygroups allowed per project
quota_driver = *cinder.quota.DbQuotaDriver*	(StrOpt) Default driver to use for quota checks
quota_gigabytes = *1000*	(IntOpt) Total amount of storage, in gigabytes, allowed for volumes and snapshots 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) Enables or disables use of default quota class with default quota.

Table 2.75. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.

Configuration option = Default value	Description
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.
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_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 2.76. Description of Redis configuration options

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 2.77. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
volume_topic = *cinder-volume*	(StrOpt) The topic that volume nodes listen on
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP

Configuration option = Default value	Description
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 2.78. Description of SAN configuration options

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_secondary_ip = *None*	(StrOpt) VNX secondary SP IP Address.
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 2.79. Description of scheduler configuration options

Configuration option = Default value	Description
[DEFAULT]	
filter_function = *None*	(StrOpt) String representation for an equation that will be used to filter hosts. Only used when the driver filter is set to be used by the Cinder scheduler.
goodness_function = *None*	(StrOpt) String representation for an equation that will be used to determine the goodness of a host. Only used when using the goodness weigher is set to be used by the Cinder scheduler.
scheduler_default_filters = *AvailabilityZoneFilter, CapacityFilter, CapabilitiesFilter*	(ListOpt) Which filter class names to use for filtering hosts when not specified in the request.

Configuration option = Default value	Description
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 that scheduler nodes listen on

Table 2.80. Description of SCST volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
scst_target_driver = *iscsi*	(StrOpt) SCST target implementation can choose from multiple SCST target drivers.
scst_target_iqn_name = *None*	(StrOpt) Certain ISCSI targets have predefined target names, SCST target driver uses this name.

Table 2.81. Description of Scality REST Block storage driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
srb_base_urls = *None*	(StrOpt) Comma-separated list of REST servers IP to connect to. (eg http://IP1/,http://IP2:81/path

Table 2.82. Description of storage configuration options

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. tgtadm is default, use lioadm for LIO iSCSI support, scstadmin for SCST target support, iseradm for the ISER protocol, ietadm for iSCSI Enterprise Target, iscsictl for Chelsio iSCSI Target or fake for testing.
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 = *None*	(IntOpt) This option is deprecated and unused. It will be removed in the Liberty release.

Configuration option = Default value	Description
iscsi_port = *3260*	(IntOpt) The port that the iSCSI daemon is listening on
iscsi_protocol = *iscsi*	(StrOpt) Determines the iSCSI protocol for new iSCSI volumes, created with tgtadm or lioadm target helpers. In order to enable RDMA, this parameter should be set with the value "iser". The supported iSCSI protocol values are "iscsi" and "iser".
iscsi_target_prefix = *iqn.2010-10.org.openstack:*	(StrOpt) Prefix for iSCSI volumes
iscsi_write_cache = *on*	(StrOpt) Sets the behavior of the iSCSI target to either perform write-back(on) or write-through(off). This parameter is valid if iscsi_helper is set to tgtadm or iseradm.
iser_helper = *tgtadm*	(StrOpt) The name of the 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 = *None*	(IntOpt) This option is deprecated and unused. It will be removed in the Liberty release.
iser_port = *3260*	(IntOpt) The port that the iSER daemon is listening on
iser_target_prefix = *iqn.2010-10.org.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.
migration_create_volume_timeout_secs = *300*	(IntOpt) Timeout for creating the volume to migrate to when performing volume migration (seconds)
num_iser_scan_tries = *3*	(IntOpt) The maximum number of times to rescan iSER targetto 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
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_copy_blkio_cgroup_name = *cinder-volume-copy*	(StrOpt) The blkio cgroup name to be used to limit bandwidth of volume copy
volume_copy_bps_limit = *0*	(IntOpt) The upper limit of bandwidth of volume copy. 0 => unlimited
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 for which to generate volume usages. The options are hour, day, month, or year.
volumes_dir = *$state_path/volumes*	(StrOpt) Volume configuration file storage directory

Table 2.83. Description of Violin volume driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
gateway_mga = *None*	(StrOpt) IP address or hostname of mg-a
gateway_mgb = *None*	(StrOpt) IP address or hostname of mg-b
use_igroups = *False*	(BoolOpt) Use igroups to manage targets and initiators

Table 2.84. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 2.85. Description of zones configuration options

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

New, updated and deprecated options in Kilo for OpenStack Block Storage

Table 2.86. New options

Option = default value	(Type) Help string
[DEFAULT] backend_host = None	(StrOpt) Backend override of host value.
[DEFAULT] backup_container = None	(StrOpt) Custom container to use for backups.
[DEFAULT] backup_enable_progress_timer = True	(BoolOpt) Enable or Disable the timer to send the periodic progress notifications to Ceilometer when backing up the volume to the backend storage. The default value is True to enable the timer.
[DEFAULT] backup_file_size = 1999994880	(IntOpt) The maximum size in bytes of the files used to hold backups. If the volume being backed up exceeds this size, then it will be backed up into multiple files.
[DEFAULT] backup_mount_options = None	(StrOpt) Mount options passed to the NFS client. See NFS man page for details.
[DEFAULT] backup_mount_point_base = $state_path/backup_mount	(StrOpt) Base dir containing mount point for NFS share.

Option = default value	(Type) Help string
[DEFAULT] backup_object_number_per_notification = 10	(IntOpt) The number of chunks or objects, for which one Ceilometer notification will be sent
[DEFAULT] backup_sha_block_size_bytes = 32768	(IntOpt) The size in bytes that changes are tracked for incremental backups. backup_swift_object_size has to be multiple of backup_swift_block_size.
[DEFAULT] backup_share = None	(StrOpt) NFS share in fqdn:path, ipv4addr:path, or "[ipv6addr]:path" format.
[DEFAULT] backup_swift_block_size = 32768	(IntOpt) The size in bytes that changes are tracked for incremental backups. backup_swift_object_size has to be multiple of backup_swift_block_size.
[DEFAULT] backup_swift_enable_progress_timer = True	(BoolOpt) Enable or Disable the timer to send the periodic progress notifications to Ceilometer when backing up the volume to the Swift backend storage. The default value is True to enable the timer.
[DEFAULT] backup_timer_interval = 120	(IntOpt) Interval, in seconds, between two progress notifications reporting the backup status
[DEFAULT] cb_account_name = None	(StrOpt) CloudByte storage specific account name. This maps to a project name in OpenStack.
[DEFAULT] cb_add_qosgroup = {'latency': '15', 'iops': '10', 'graceallowed': 'false', 'iopscontrol': 'true', 'memlimit': '0', 'throughput': '0', 'tpcontrol': 'false', 'networkspeed': '0'}	(DictOpt) These values will be used for CloudByte storage's addQos API call.
[DEFAULT] cb_apikey = None	(StrOpt) Driver will use this API key to authenticate against the CloudByte storage's management interface.
[DEFAULT] cb_confirm_volume_create_retries = 3	(IntOpt) Will confirm a successful volume creation in CloudByte storage by making this many number of attempts.
[DEFAULT] cb_confirm_volume_create_retry_interval = 5	(IntOpt) A retry value in seconds. Will be used by the driver to check if volume creation was successful in CloudByte storage.
[DEFAULT] cb_create_volume = {'compression': 'off', 'deduplication': 'off', 'blocklength': '512B', 'sync': 'always', 'protocoltype': 'ISCSI', 'recordsize': '16k'}	(DictOpt) These values will be used for CloudByte storage's createVolume API call.
[DEFAULT] cb_tsm_name = None	(StrOpt) This corresponds to the name of Tenant Storage Machine (TSM) in CloudByte storage. A volume will be created in this TSM.
[DEFAULT] chap_password =	(StrOpt) Password for specified CHAP account name.
[DEFAULT] chap_username =	(StrOpt) CHAP user name.
[DEFAULT] check_max_pool_luns_threshold = False	(BoolOpt) Report free_capacity_gb as 0 when the limit to maximum number of pool LUNs is reached. By default, the value is False.
[DEFAULT] chiscsi_conf = /etc/chelsio-iscsi/chiscsi.conf	(StrOpt) Chiscsi (CXT) global defaults configuration file
[DEFAULT] dell_sc_api_port = 3033	(IntOpt) Dell API port
[DEFAULT] dell_sc_server_folder = openstack	(StrOpt) Name of the server folder to use on the Storage Center
[DEFAULT] dell_sc_ssn = 64702	(IntOpt) Storage Center System Serial Number
[DEFAULT] dell_sc_volume_folder = openstack	(StrOpt) Name of the volume folder to use on the Storage Center
[DEFAULT] driver_data_namespace = None	(StrOpt) Namespace for driver private data values to be saved in.
[DEFAULT] driver_use_ssl = False	(BoolOpt) Tell driver to use SSL for connection to backend storage if the driver supports it.
[DEFAULT] end_time = None	(StrOpt) If this option is specified then the end time specified is used instead of the end time of the last completed audit period.

Option = default value	(Type) Help string
[DEFAULT] enforce_multipath_for_image_xfer = False	(BoolOpt) If this is set to True, attachment of volumes for image transfer will be aborted when multipathd is not running. Otherwise, it will fallback to single path.
[DEFAULT] filter_function = None	(StrOpt) String representation for an equation that will be used to filter hosts. Only used when the driver filter is set to be used by the Cinder scheduler.
[DEFAULT] flashsystem_connection_protocol = FC	(StrOpt) Connection protocol should be FC.
[DEFAULT] flashsystem_multihostmap_enabled = True	(BoolOpt) Allows vdisk to multi host mapping.
[DEFAULT] flashsystem_multipath_enabled = False	(BoolOpt) Connect with multipath (FC only).
[DEFAULT] force_delete_lun_in_storagegroup = False	(BoolOpt) Delete a LUN even if it is in Storage Groups.
[DEFAULT] gateway_mga = None	(StrOpt) IP address or hostname of mg-a
[DEFAULT] gateway_mgb = None	(StrOpt) IP address or hostname of mg-b
[DEFAULT] goodness_function = None	(StrOpt) String representation for an equation that will be used to determine the goodness of a host. Only used when using the goodness weigher is set to be used by the Cinder scheduler.
[DEFAULT] initiator_auto_deregistration = False	(BoolOpt) Automatically deregister initiators after the related storage group is destroyed. By default, the value is False.
[DEFAULT] iscsi_protocol = iscsi	(StrOpt) Determines the iSCSI protocol for new iSCSI volumes, created with tgtadm or lioadm target helpers. In order to enable RDMA, this parameter should be set with the value "iser". The supported iSCSI protocol values are "iscsi" and "iser".
[DEFAULT] iscsi_secondary_ip_addresses =	(ListOpt) The list of secondary IP addresses of the iSCSI daemon
[DEFAULT] ise_completion_retries = 30	(IntOpt) Number on retries to get completion status after issuing a command to ISE.
[DEFAULT] ise_connection_retries = 5	(IntOpt) Number of retries (per port) when establishing connection to ISE management port.
[DEFAULT] ise_raid = 1	(IntOpt) Raid level for ISE volumes.
[DEFAULT] ise_retry_interval = 1	(IntOpt) Interval (secs) between retries.
[DEFAULT] ise_storage_pool = 1	(IntOpt) Default storage pool for volumes.
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative – log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] lvm_conf_file = /etc/cinder/lvm.conf	(StrOpt) LVM conf file to use for the LVM driver in Cinder; this setting is ignored if the specified file does not exist (You can also specify 'None' to not use a conf file even if one exists).
[DEFAULT] max_over_subscription_ratio = 20.0	(FloatOpt) Float representation of the over subscription ratio when thin provisioning is involved. Default ratio is 20.0, meaning provisioned capacity can be 20 times of the total

Option = default value	(Type) Help string
	physical capacity. If the ratio is 10.5, it means provisioned capacity can be 10.5 times of the total physical capacity. A ratio of 1.0 means provisioned capacity cannot exceed the total physical capacity. A ratio lower than 1.0 will be ignored and the default value will be used instead.
[DEFAULT] nas_mount_options = None	(StrOpt) Options used to mount the storage backend file system where Cinder volumes are stored.
[DEFAULT] nas_secure_file_operations = auto	(StrOpt) Allow network-attached storage systems to operate in a secure environment where root level access is not permitted. If set to False, access is as the root user and insecure. If set to True, access is not as root. If set to auto, a check is done to determine if this is a new installation: True is used if so, otherwise False. Default is auto.
[DEFAULT] nas_secure_file_permissions = auto	(StrOpt) Set more secure file permissions on network-attached storage volume files to restrict broad other/world access. If set to False, volumes are created with open permissions. If set to True, volumes are created with permissions for the cinder user and group (660). If set to auto, a check is done to determine if this is a new installation: True is used if so, otherwise False. Default is auto.
[DEFAULT] nas_share_path =	(StrOpt) Path to the share to use for storing Cinder volumes. For example: "/srv/export1" for an NFS server export available at 10.0.5.10:/srv/export1 .
[DEFAULT] netapp_partner_backend_name = None	(StrOpt) The name of the config.conf stanza for a Data ONTAP (7-mode) HA partner. 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 it is required if the storage protocol selected is FC.
[DEFAULT] nfs_mount_attempts = 3	(IntOpt) The number of attempts to mount nfs shares before raising an error. At least one attempt will be made to mount an nfs share, regardless of the value specified.
[DEFAULT] os_privileged_user_name = None	(StrOpt) OpenStack privileged account username. Used for requests to other services (such as Nova) that require an account with special rights.
[DEFAULT] os_privileged_user_password = None	(StrOpt) Password associated with the OpenStack privileged account.
[DEFAULT] os_privileged_user_tenant = None	(StrOpt) Tenant name associated with the OpenStack privileged account.
[DEFAULT] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[DEFAULT] public_endpoint = None	(StrOpt) Public url to use for versions endpoint. The default is None, which will use the request's host_url attribute to populate the URL base. If Cinder is operating behind a proxy, you will want to change this to represent the proxy's URL.
[DEFAULT] quobyte_client_cfg = None	(StrOpt) Path to a Quobyte Client configuration file.
[DEFAULT] quobyte_mount_point_base = $state_path/mnt	(StrOpt) Base dir containing the mount point for the Quobyte volume.
[DEFAULT] quobyte_qcow2_volumes = True	(BoolOpt) Create volumes as QCOW2 files rather than raw files.
[DEFAULT] quobyte_sparsed_volumes = True	(BoolOpt) Create volumes as sparse 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.

Option = default value	(Type) Help string
[DEFAULT] quobyte_volume_url = None	(StrOpt) URL to the Quobyte volume e.g., quobyte://<DIR host>/<volume name>
[DEFAULT] request_timeout = 300	(IntOpt) Global backend request timeout, in seconds
[DEFAULT] scst_target_driver = iscsi	(StrOpt) SCST target implementation can choose from multiple SCST target drivers.
[DEFAULT] scst_target_iqn_name = None	(StrOpt) Certain ISCSI targets have predefined target names, SCST target driver uses this name.
[DEFAULT] send_actions = False	(BoolOpt) Send the volume and snapshot create and delete notifications generated in the specified period.
[DEFAULT] sf_allow_template_caching = True	(BoolOpt) Create an internal cache of copy of images when a bootable volume is created to eliminate fetch from glance and qemu-conversion on subsequent calls.
[DEFAULT] sf_template_account_name = openstack-vtemplate	(StrOpt) Account name on the SolidFire Cluster to use as owner of template/cache volumes (created if does not exist).
[DEFAULT] srb_base_urls = None	(StrOpt) Comma-separated list of REST servers IP to connect to. (eg http://IP1/,http://IP2:81/path
[DEFAULT] start_time = None	(StrOpt) If this option is specified then the start time specified is used instead of the start time of the last completed audit period.
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
[DEFAULT] use_chap_auth = False	(BoolOpt) Option to enable/disable CHAP authentication for targets.
[DEFAULT] use_igroups = False	(BoolOpt) Use igroups to manage targets and initiators
[DEFAULT] vpool_name =	(StrOpt) Vpool to use for volumes - backend is defined by vpool not by us.
[DEFAULT] xtremio_cluster_name =	(StrOpt) XMS cluster id in multi-cluster environment
[DEFAULT] zfssa_data_ip = None	(StrOpt) Data path IP address
[DEFAULT] zfssa_https_port = 443	(StrOpt) HTTPS port number
[DEFAULT] zfssa_nfs_mount_options =	(StrOpt) Options to be passed while mounting share over nfs
[DEFAULT] zfssa_nfs_pool =	(StrOpt) Storage pool name.
[DEFAULT] zfssa_nfs_project = NFSProject	(StrOpt) Project name.
[DEFAULT] zfssa_nfs_share = nfs_share	(StrOpt) Share name.
[DEFAULT] zfssa_nfs_share_compression = off	(StrOpt) Data compression.
[DEFAULT] zfssa_nfs_share_logbias = latency	(StrOpt) Synchronous write bias-latency, throughput.
[oslo_concurrency] disable_process_locking = False	(BoolOpt) Enables or disables inter-process locks.
[oslo_concurrency] lock_path = None	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP

Option = default value	(Type) Help string
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.

Table 2.87. New default values

Option	Previous default value	New default value
[DEFAULT] backup_metadata_version	1	2
[DEFAULT] client_socket_timeout	0	900
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] iscsi_num_targets	100	None
[DEFAULT] iser_num_targets	100	None
[DEFAULT] iser_target_prefix	iqn.2010-10.org.iser.openstack:	iqn.2010-10.org.openstack:

Option	Previous default value	New default value
[DEFAULT] nova_catalog_admin_info	compute:nova:adminURL	compute:Compute Service:adminURL
[DEFAULT] nova_catalog_info	compute:nova:publicURL	compute:Compute Service:publicURL
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	oslo_messaging._drivers.matchmaker.MatchMakerLocalhost
[keymgr] encryption_auth_url	http://localhost:5000/v2.0	http://localhost:5000/v3

Table 2.88. Deprecated options

Deprecated option	New Option
[DEFAULT] log-format	None
[DEFAULT] use-syslog	None
[DEFAULT] use_syslog	None
[DEFAULT] osapi_max_request_body_size	[oslo_middleware] max_request_body_size
[DEFAULT] eqlx_chap_password	[DEFAULT] chap_password
[DEFAULT] datera_api_token	None
[DEFAULT] eqlx_use_chap	[DEFAULT] use_chap_auth
[DEFAULT] enable_v1_api	None
[DEFAULT] db_backend	[database] backend
[DEFAULT] host	[DEFAULT] backend_host
[DEFAULT] eqlx_chap_login	[DEFAULT] chap_username
[DEFAULT] log_format	None

3. Compute

Table of Contents

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" [xxi].

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 3.20, "Description of common configuration options" [289].

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.
[baremetal]	Configures the baremetal hypervisor driver.
[cells]	Configures cells functionality. For details, see the section called "Cells" [274].
[conductor]	Configures the `nova-conductor` service.
[database]	Configures the database that Compute uses.
[glance]	Configures how to access the Image service.
[hyperv]	Configures the Hyper-V hypervisor driver.
[image_file_url]	Configures additional filesystems to access the Image Service.
[keymgr]	Configures the key manager.
[keystone_authtoken]	Configures authorization via Identity service.
[libvirt]	Configures the hypervisor drivers using the Libvirt library: KVM, LXC, Qemu, UML, Xen.
[matchmaker_redis]	Configures a Redis server.
[matchmaker_ring]	Configures a matchmaker ring.
[metrics]	Configures weights for the metrics weighter.
[neutron]	Configures Networking specific options.
[osapi_v3]	Configures the OpenStack Compute API v3.
[rdp]	Configures RDP proxying.
[serial_console]	Configures serial console.

[spice]	Configures virtual consoles using SPICE.
[ssl]	Configures certificate authority using SSL.
[trusted_computing]	Configures the trusted computing pools functionality and how to connect to a remote attestation service.
[upgrade_levels]	Configures version locking on the RPC (message queue) communications between the various Compute services to allow live upgrading an OpenStack installation.
[vmware]	Configures the VMware hypervisor driver.
[xenserver]	Configures the XenServer hypervisor driver.
[zookeeper]	Configures the ZooKeeper ServiceGroup driver.

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 3.39, "Description of logging configuration options" [300].

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 3.14, "Description of authentication configuration options" [286].

To customize certificate authority settings for Compute, use the configuration options documented in Table 3.18, "Description of CA and SSL configuration options" [288].

To customize Compute and the Identity service to use LDAP as a backend, refer to the configuration options documented in Table 3.36, "Description of LDAP configuration options" [298].

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 ac-

cess 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 3.25, "Description of database configuration options" [293] 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 `rabbit`.

```
rpc_backend=rabbit
```

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.openstack.common.notifier.rpc_notifier` in the `nova.conf` file. The de-

fault for sending usage data is sixty seconds plus a random number of seconds from zero to sixty.

Table 3.1. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
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_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

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=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`* parameter 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 3.2. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
`amqp_auto_delete` = *`False`*	(BoolOpt) Auto-delete queues in AMQP.
`amqp_durable_queues` = *`False`*	(BoolOpt) Use durable queues in AMQP.
`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_receiver_capacity` = *`1`*	(IntOpt) The number of prefetched messages held by receiver.
`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.

Configuration option = Default value	Description
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

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 3.3. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 3.4. Description of AMQP configuration options

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.
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.
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 3.5. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
matchmaker_heartbeat_freq = *300*	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = *600*	(IntOpt) Heartbeat time-to-live.

Configuration option = Default value	Description
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread 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.
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout
[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 403 Forbidden.

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 3.6. 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

HTTP method	API URI	API regular expression	Limit
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 3.12, “Description of API configuration options” [284].

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 3.27, "Description of EC2 configuration options" [295].

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 support Fibre Channel automatic zoning on Brocade and Cisco switches. On other hardware 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 or 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
```

iSCSI interface and offload support in Compute

Note

iSCSI interface and offload support is only present since Kilo.

Compute supports open-iscsi iSCSI interfaces for offload cards. Offload hardware must be present and configured on every compute node where offload is desired. Once an open-iscsi interface is configured, the iface name (`iface.iscsi_ifacename`) should be passed to libvirt via the `iscsi_iface` parameter for use. All iscsi sessions will be bound to this iSCSI interface.

Currently supported transports (iface.transport_name) are be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx, ocs. No configuration changes are needed outside of Compute node.

iSER is currently supported via the separate iSER LibvirtISERVolumeDriver and will be rejected if used via the `iscsi_iface` parameter.

iSCSI iface configuration

- Note the distinction between the transport name (`iface.transport_name`) and iface name (`iface.iscsi_ifacename`). The actual iface name must be specified via the `iscsi_iface` parameter to libvirt for offload to work.
- The default name for an iscsi iface (open-iscsi parameter `iface.iscsi_ifacename`) is in the format transport_name.hwaddress when generated by iscsiadm.
- iscsiadm can be used to view and generate current iface configuration. Every network interface that supports an open-iscsi transport can have one or more iscsi ifaces associated with it. If no ifaces have been configured for a network interface supported by an open-iscsi transport, this command will create a default iface configuration for that network interface. For example :

```
# iscsiadm -m iface
   default tcp,<empty>,<empty>,<empty>,<empty>
   iser iser,<empty>,<empty>,<empty>,<empty>
   bnx2i.00:05:b5:d2:a0:c2 bnx2i,00:05:b5:d2:a0:c2,5.10.10.20,<empty>,
<empty>
   cxgb4i.00:07:43:28:b2:58 cxgb4i,00:07:43:28:b2:58,102.50.50.80,<empty>,
<empty>
   qla4xxx.00:c0:dd:08:63:ea qla4xxx,00:c0:dd:08:63:ea,20.15.0.9,<empty>,
<empty>
```

  The output is in the format : *iface_name transport_name,hwaddress,ipaddress,net_ifacename,initiatorname.*

- Individual iface configuration can be viewed via

```
# iscsiadm -m iface -I IFACE_NAME
   # BEGIN RECORD 2.0-873
   iface.iscsi_ifacename = cxgb4i.00:07:43:28:b2:58
   iface.net_ifacename = <empty>
   iface.ipaddress = 102.50.50.80
   iface.hwaddress = 00:07:43:28:b2:58
   iface.transport_name = cxgb4i
   iface.initiatorname = <empty>
   # END RECORD
```

  Configuration can be updated as desired via

```
# iscsiadm -m iface-I IFACE_NAME--op=update -n iface.SETTING -v VALUE
```

- All iface configurations need a minimum of `iface.iface_name`, `iface.transport_name` and `iface.hwaddress` to be correctly configured to work. Some transports may require `iface.ipaddress` and `iface.net_ifacename` as well to bind correctly.

Detailed configuration instructions can be found here http://www.open-iscsi.org/docs/README

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" [261] and the section called "ImagePropertiesFilter" [264] to schedule different hypervisors within the same installation. The following links help you choose a hypervisor. See http://docs.openstack.org/developer/nova/support-matrix.html 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), used 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 (using libvirt) - Xen Project Hypervisor using libvirt as management interface into `nova-compute` to run Linux, Windows, FreeBSD and NetBSD virtual machines.
- XenServer - XenServer, Xen Cloud Platform (XCP) and other XAPI based Xen variants runs 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.

Hypervisor configuration basics

The node where the `nova-compute` service is installed and operates on the same node that runs all of the virtual machines. This is 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 3.32, "Description of hypervisor configuration options" [297].

Specific options for particular hypervisors can be found in the 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 Guide*.

Enable KVM

The following sections outline how to enable KVM based hardware virtualisation on different architectures and platforms. To perform these steps, you must be logged in as the `root` user.

For x86 based systems

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 capable of hardware-virtualization. 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`.

2. 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
```

For POWER based systems

KVM as a hypervisor is supported on POWER system's PowerNV platform.

1. To determine if your POWER platform supports KVM based virtualization run the following command:

   ```
   #cat /proc/cpuinfo | grep PowerNV
   ```

   If the previous command generates the following output, then CPU supports KVM based virtualization

   ```
   platform: PowerNV
   ```

   If no output is displayed, then your POWER platform does not support KVM based hardware virtualization.

2. To list the loaded kernel modules and verify that the `kvm` modules are loaded, run the following command:

   ```
   # lsmod | grep kvm
   ```

   If the output includes `kvm_hv`, 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 the following command to load it:

```
# modprobe -a kvm
```

For PowerNV platform, run the following command:

```
# modprobe -a kvm-hv
```

Because a KVM installation can change user group membership, you might need to log in again for changes to take effect.

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 the 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 a matching host CPU.

Custom

If your `nova.conf` file contains `cpu_mode=custom`, you can explicitly specify one of the supported named models 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 the 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 ERROR state, 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 set correctly. 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)

XenServer (and other XAPI based Xen variants)

This section describes XAPI managed hypervisors, and how to use them with OpenStack.

Terminology

Xen

A hypervisor that provides the fundamental isolation between virtual machines. Xen is open source (GPLv2) and is managed by XenProject.org, a cross-industry organization and a Linux Foundation Collaborative project.

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 toolstack you are using. Make sure you know what toolstack you want before you get started.

XAPI

XAPI is one of the toolstacks that could control a Xen based hypervisor. XAPI's role is similar to libvirt's in the KVM world. The API provided by XAPI is called XenAPI. To learn more about the provided interface, look at XenAPI Object Model Overview for definitions of XAPI specific terms such as SR, VDI, VIF and PIF.

OpenStack has a compute driver which talks to XAPI, therefore all XAPI managed servers could be used with OpenStack.

XenAPI

XenAPI is the API provided by XAPI. This name is also used by the python library that is a client for XAPI.

XenServer

An Open Source virtualization platform that delivers all features needed for any server and datacenter implementation including the Xen hypervisor and XAPI for the management. For more information and product downloads, visit xenserver.org .

XCP

XCP is not supported anymore. XCP project recommends all XCP users to upgrade to the latest version of XenServer by visiting xenserver.org .

XenServer-core

This is a method for building the core packages in a XenServer installation on an existing RPM-based system. Initial support for this configuration (notably running Compute services in domain 0) was added in Havana. XenServer-core for Debian/Ubuntu is built from the main branch and, therefore, is continuously up to date.

Kronos

This is a project initiated to provide the ability to install XAPI toolstack onto an existing Debian-based deployment. For more information, visit the Xen wiki wiki.xenproject.org/wiki/Project_Kronos .

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 is known as a domU or guest. All customer VMs are unprivileged, but you should note that on Xen, the OpenStack Compute service (`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.

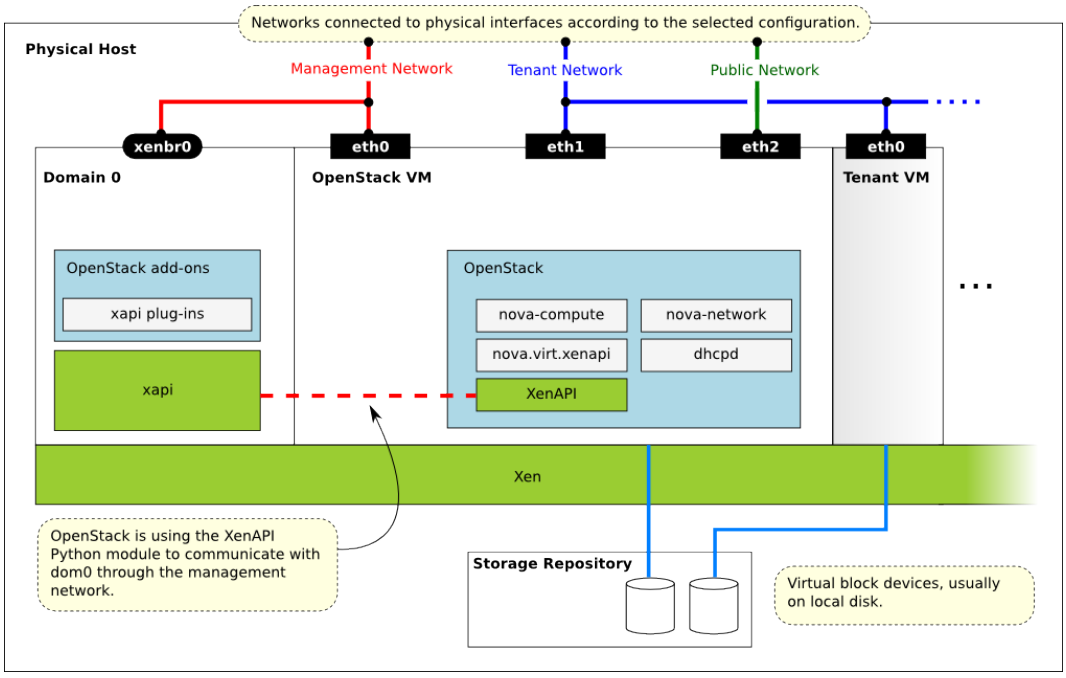
Paravirtualized versus hardware virtualized domains

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

A basic OpenStack deployment on a XAPI-managed server, assuming that the network provider is nova-network, looks like this:



Key things to note:

- The hypervisor: Xen
- Domain 0: runs XAPI and some small pieces from OpenStack, the XAPI plug-ins.
- OpenStack VM: The `Compute` service runs in a paravirtualized virtual machine, on the host under management. Each host runs a local instance of `Compute`. It is also running an instance of nova-network.

- OpenStack Compute uses the XenAPI Python library to talk to XAPI, and it uses the Management Network to reach from the OpenStack VM to Domain 0.

Some notes on the networking:

- The above diagram assumes FlatDHCP networking.
- There are three main OpenStack networks:
  - Management network: RabbitMQ, MySQL, inter-host communication, and compute-XAPI communication. Please note that the VM images are downloaded by the XenAPI plug-ins, so make sure that the OpenStack Image service is accessible through this network. It usually means binding those services to the management interface.
  - Tenant network: controlled by nova-network, this is used for tenant traffic.
  - 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.)

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 XenProject.org: XenProject.org > Users > Cloud
- Xen Hypervisor project: http://www.xenproject.org/developers/teams/hypervisor.html
- Xapi project: http://www.xenproject.org/developers/teams/xapi.html
- Further XenServer and OpenStack information: http://wiki.openstack.org/XenServer

Install XenServer

Before you can run OpenStack with XenServer, you must install the hypervisor 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 before you install the operating system where you want to run OpenStack code. You then install `nova-compute` into a dedicated virtual machine on the host.

Use the following link to download XenServer's installation media:

- http://xenserver.org/open-source-virtualization-download.html

When you install many servers, you might find it easier to perform PXE boot installations . You can also package any post-installation changes that you want to make to your XenServer by following the instructions of creating your own XenServer supplemental pack .

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 XAPI-specific term relating to the physical storage where virtual disks are stored.

On the XenServer 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

The following steps need to be completed after the hypervisor's installation:

1. For resize and migrate functionality, enable password-less SSH authentication and set up the `/images` directory on dom0.
2. Install the XAPI plug-ins.
3. To support AMI type images, you must set up `/boot/guest` symlink/directory in dom0.
4. Create a paravirtualized virtual machine that can run `nova-compute`.
5. Install and configure `nova-compute` in the above virtual machine.

Install XAPI plug-ins

When you use a XAPI managed hypervisor, you can install a Python script (or any executable) on the host side, and execute that through XenAPI. These scripts are called plug-ins. The OpenStack related XAPI plug-ins live in OpenStack Compute's code repository. These plug-ins have to be copied to dom0's filesystem, to the appropriate directory, where XAPI can find them. It is important to ensure that the version of the plug-ins are in line with the OpenStack Compute installation you are using.

The plugins should typically be copied from the Nova installation running in the Compute's DomU, but if you want to download the latest version the following procedure can be used.

Manually installing the plug-ins

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, and the XenServer host is accessible as xenserver. Match those parameters to your setup.

```
$ NOVA_URL=https://github.com/openstack/nova/archive/master.zip
$ wget -qO "$NOVA_ZIPBALL" "$NOVA_URL"
$ unzip "$NOVA_ZIPBALL" -d "$NOVA_SOURCES"
```

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/
```

4. Remove temporary files/directories:

```
$ rm "$NOVA_ZIPBALL"
$ rm -rf "$NOVA_SOURCES"
```

Prepare for AMI type images

To support AMI type images in your OpenStack installation, you must create the `/boot/guest` directory on dom0. One of the OpenStack XAPI plugins will extract the kernel and ramdisk from AKI and ARI images and put them to that directory.

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 up dom0's filesystem, 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 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 a `/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 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
```

XenAPI configuration reference

The following section discusses some commonly changed options when using the XenAPI driver. The table below provides a complete reference of all configuration options available for configuring XAPI with OpenStack.

The recommended way to use XAPI with OpenStack is through the XenAPI driver. To enable the XenAPI driver, add the following configuration options to `/etc/nova/nova.conf` and restart `OpenStack Compute`:

```
compute_driver = xenapi.XenAPIDriver
[xenserver]
connection_url = http://your_xenapi_management_ip_address
connection_username = root
connection_password = your_password
```

These connection details are used by OpenStack Compute service to contact your hypervisor and are the same details you use to connect XenCenter, the XenServer management console, to your XenServer node.

Note

The `connection_url` is generally the management network IP address of the XenServer.

Agent

The agent is a piece of software that runs on the instances, and communicates with OpenStack. In case of the XenAPI driver, the agent communicates with OpenStack through XenStore (see the Xen Project Wiki for more information on XenStore).

If you don't have the guest agent on your VMs, it takes a long time for OpenStack Compute to detect that the VM has successfully started. Generally a large timeout is required for Windows instances, but you may want to adjust: `agent_version_timeout` within the `[xenserver]` section.

VNC proxy address

Assuming you are talking to XAPI through a management network, and XenServer is on the address: 10.10.1.34 specify the same address for the vnc proxy address: `vncserver_proxyclient_address=10.10.1.34`

Storage

You can specify which Storage Repository to use with nova by editing the following flag. To use the local-storage setup by the default installer:

```
sr_matching_filter = "other-config:i18n-key=local-storage"
```

Another 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"
```

XenAPI configuration reference

To customize the XenAPI driver, use the configuration option settings documented in Table 3.62, "Description of Xen configuration options" [315].

Xen via Libvirt

OpenStack Compute supports the Xen Project Hypervisor (or Xen). Xen can be integrated with OpenStack Compute via the libvirt toolstack or via the XAPI toolstack. This document describes how to set up OpenStack Compute with Xen and libvirt. For information on how to set up Xen with XAPI refer to the section called "XenServer (and other XAPI based Xen variants)" [229].

Installing Xen with Libvirt

At this stage we recommend to use the baseline that we use for the Xen Project OpenStack CI Loop, which contains the most recent stability fixes to both Xen and Libvirt.

Xen 4.5.1 (or newer) and Libvirt 1.2.15 (or newer) contain the most recent OpenStack improvements for Xen. The necessary Xen changes have also been backported to the Xen 4.4.3 stable branch (not yet released at this stage). Please check with the Linux and FreeBSD distros you are intending to use as Dom 0, whether the relevant version of Xen and Libvirt are available as installable packages.

Alternatively, it is possible to use the Ubuntu LTS 14.04 Xen Package *4.4.1-0ubuntu0.14.04.4* (Xen 4.4.1) and apply the patches outlined here. You can also use the Ubuntu LTS 14.04 libvirt package *1.2.2 libvirt_1.2.2-0ubuntu13.1.7* as baseline and update it to libvirt version 1.2.15, or 1.2.14 with the patches outlined here applied. Note that this will require re-build these packages partly from source.

For further information and latest developments, you may want to consult the Xen Project's mailing lists for OpenStack related issues and questions.

Configuring Xen with Libvirt

To enable Xen via libvirt, 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 = xen
```

Additional Configuration Options

Use the following as a guideline for configuring Xen for use in OpenStack:

1. **Dom0 Memory:** Set it between 1GB and 4GB by adding the following parameter to the `grub.conf` file.

   ```
   dom0_mem=1024M
   ```

Note

Note that the location of the grub.conf file depends on the host Linux distribution that you are using. Please refer to the distro documentation for more details (see Dom 0 for more resources).

2. **Dom0 vcpus:** Set the virtual CPUs to 4 and employ CPU pinning by adding the following parameters to the `grub.conf` file.

```
dom0_max_vcpus=4 dom0_vcpus_pin
```

3. **PV vs HVM guests:** A Xen virtual machine can be paravirtualized (PV) or hardware virtualized (HVM). The virtualization mode determines the interaction between Xen, Dom 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 Dom 0 and the choice of virtualization mode determines performance characteristics. For an overview of Xen virtualization modes, see Xen Guest Types.

   In OpenStack, customer VMs may run in either PV or HVM mode. However, Dom 0 (that's the VM which is running nova-compute) must be running in PV mode. Use the following OpenStack nova commands to select the desired Xen virtualization mode

   To choose one of the HVM modes (HVM, HVM with PV Drivers or PVHVM) use

```
# nova image-meta set vm_mode=HVM
```

   To chose PV mode, which is supported by NetBSD, FreeBSD and Linux, use

```
# nova image-meta set vm_mode=XEN
```

Note

The default for virtualization mode in nova is PV mode.

To customize the libvirt driver, use the configuration option settings documented in Table 3.62, “Description of Xen configuration options” [315].

Known Issues

- Xen via libvirt is currently only supported with nova networking. A number of bugs are currently worked on to make sure that Xen via libvirt will also work with Neutron.

Additional Information and Resources

The following section contains links to other useful resources

- wiki.xenproject.org/wiki/OpenStack - OpenStack Documentation on the Xen Project wiki
- wiki.xenproject.org/wiki/OpenStack_CI_Loop_for_Xen-Libvirt - Information about the Xen Project OpenStack CI Loop
- wiki.xenproject.org/wiki/OpenStack_via_DevStack - How to set up OpenStack via DevStack

- Mailing lists for OpenStack related issues and questions - This list is dedicated to coordinating bug fixes and issues across Xen, libvirt and OpenStack and the CI loop.

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, 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 later are supported.

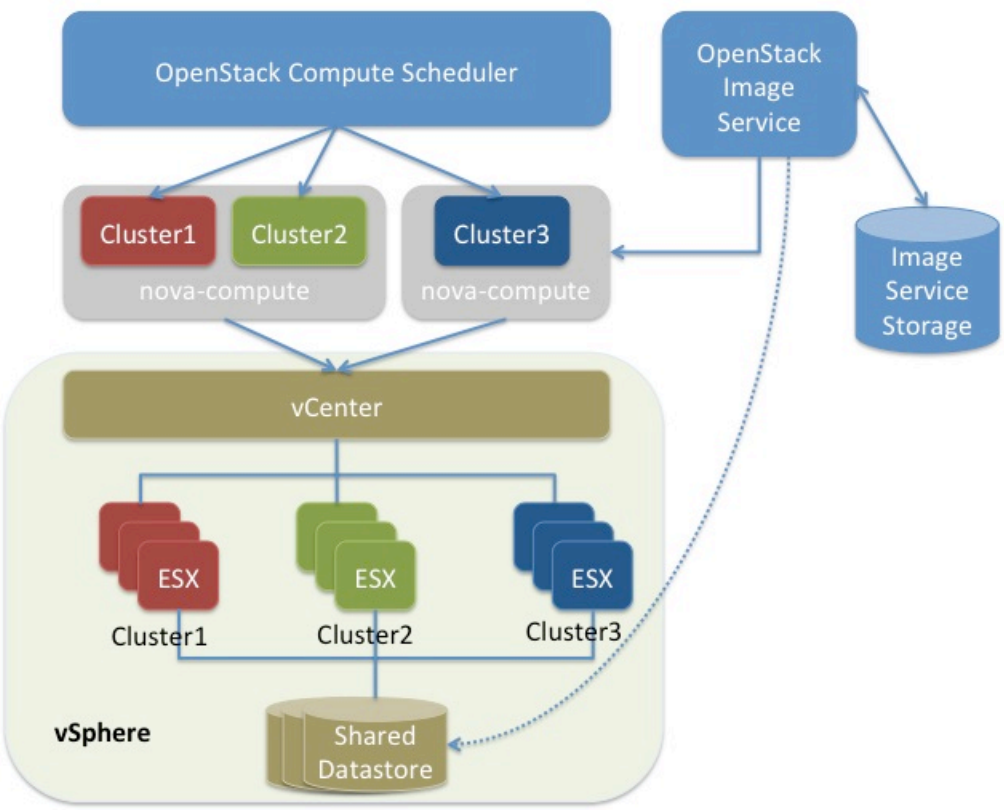
The VMware vCenter driver enables the `nova-compute` service to communicate with a VMware vCenter server that manages one or more ESX host clusters. The driver aggregates the ESX hosts in each cluster to present one large hypervisor entity for each cluster to the Compute scheduler. Because individual 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 3.1. 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" [247].

Configuration overview

To get started with the VMware vCenter driver, complete the following high-level steps:

1. Configure vCenter. See the section called "Prerequisites and limitations" [240].
2. Configure the VMware vCenter driver in the `nova.conf` file. See the section called "VMware vCenter driver" [242].
3. Load desired VMDK images into the OpenStack Image Service. See the section called "Images with VMware vSphere" [244].
4. Configure networking with either `nova-network` or the OpenStack Networking Service. See the section called "Networking with VMware vSphere" [247].

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" [247].
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.

Note

The VIB can be downloaded from https://github.com/openstack-vmwareapi-team/Tools.

8. 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 service account

OpenStack integration requires a vCenter service account with the following minimum permissions. Apply the permissions to the `Datacenter` root object, and select the **Propagate to Child Objects** option.

Table 3.7. vCenter permissions tree

All Privileges		
	Datastore	
		Allocate space
		Browse datastore
		Low level file operation
		Remove file
	Extension	
		Register extension
	Folder	
		Create folder
	Host	
		Configuration
	Network	
		Assign network
	Resource	
		Assign virtual machine to resource pool

		Migrate powered off virtual machine
		Migrate powered on virtual machine
	Virtual Machine	
		Configuration
		Interaction
		Inventory
		Provisioning
		Sessions
		Snapshot management
	vApp	
		Export
		Import

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.
- The vCenter driver generates instance name by instance ID. Instance name template is ignored.
- The minimum supported vCenter version is 5.1.0. In OpenStack Liberty release this will be logged as a warning. In OpenStack "M" release this will be enforced.

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" [248].

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 with the VMware vCenter driver. The following sections provide additional details on the supported disks and the commands used for conversion and upload.

Supported image types

Upload images to the OpenStack Image service in VMDK format. The following VMDK disk types are supported:

- *VMFS Flat Disks* (includes thin, thick, zeroedthick, and eagerzeroedthick). Note that once a VMFS thin disk is exported from VMFS to a non-VMFS location, like the OpenStack Image service, it becomes a preallocated flat disk. This impacts the transfer time from the OpenStack Image service to the data store when the full preallocated flat disk, rather than the thin disk, must be transferred.

- *Monolithic Sparse disks*. Sparse disks get imported from the OpenStack Image service into ESX as thin provisioned disks. Monolithic Sparse disks can be obtained from VMware Fusion or can be created by converting from other virtual disk formats using the `qemu-img` utility.

The following table shows the `vmware_disktype` property that applies to each of the supported VMDK disk types:

Table 3.8. OpenStack Image service disk type settings

vmware_disktype property	VMDK disk type
sparse	Monolithic Sparse
thin	VMFS flat, thin provisioned
preallocated (default)	VMFS flat, thick/zeroedthick/eagerzeroedthick

The `vmware_disktype` property is set when an image is loaded into the OpenStack Image service. For example, the following command creates a Monolithic Sparse image by setting `vmware_disktype` to `sparse`:

```
$ glance image-create --name "ubuntu-sparse" --disk-format vmdk \
--container-format bare \
--property vmware_disktype="sparse" \
--property vmware_ostype="ubuntu64Guest" < ubuntuLTS-sparse.vmdk
```

Note

Specifying `thin` does not provide any advantage over `preallocated` with the current version of the driver. Future versions might restore the thin properties of the disk after it is downloaded to a vSphere data store.

Convert and load images

Using the `qemu-img` utility, disk images in several formats (such as, qcow2) can be converted to the VMDK format.

For example, the following command can be used to convert a qcow2 Ubuntu Trusty cloud image:

```
$ qemu-img convert -f qcow2 ~/Downloads/trusty-server-cloudimg-amd64-disk1.img
 \
-O vmdk trusty-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 Ubuntu Trusty image after the `qemu-img` conversion, the command to upload the VMDK disk should be something like:

```
$ glance image-create --name trusty-cloud --is-public False \
--container-format bare --disk-format vmdk \
--property vmware_disktype="sparse" \
--property vmware_adaptertype="ide" < \
trusty-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 \
--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, lsiLogicsas, paraVirtual) 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`, `lxc`, `uml`, and `hyperv`. Note that `qemu` is used for both QEMU and KVM hypervisor types.

```
$ glance image-create --name "ubuntu-thick-scsi" --disk-format vmdk \
--container-format bare \
--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.

Use one of the following tools to pre-convert sparse disks.

vSphere CLI tools

Sometimes called the remote CLI or rCLI.

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.

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 through scp and the vmkfstools local to the ESX host can use used to perform the conversion. After you log in to the host through ssh, run this command:

```
vmkfstools -i /vmfs/volumes/datastore1/sparse.
vmdk /vmfs/volumes/datastore1/converted.vmdk
```

vmware-vdiskmanager

`vmware-vdiskmanager` is a utility that comes bundled with VMware Fusion and VMware Workstation. The following example converts a sparse disk to preallocated format:

```
'/Applications/VMware Fusion.app/Contents/
Library/vmware-vdiskmanager' -r sparse.vmdk -t 4
 converted.vmdk
```

In the previous cases, the converted vmdk is actually a pair of files:

- The descriptor file *converted.vmdk*.
- 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. A cached image is stored in a folder named after the image ID. 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" [250].

Note

You can also use the `vmware_linked_clone` property in the OpenStack Image service to override the linked_clone mode on a per-image basis.

If multiple compute nodes are running on the same host, or have a shared file system, you can enable them to use the same cache folder on the back-end data store. To configure this action, set the `cache_prefix` option in the `nova.conf` file. Its value stands for the name prefix of the folder where cached images are stored.

Note

This can take effect only if compute nodes are running on the same host, or have a shared file system.

You can automatically purge unused images after a specified period of time. To configure this action, set these options in the `DEFAULT` section in the `nova.conf` file:

`remove_unused_base_images` Set this option to **`True`** to specify that unused images should be removed after the duration specified in the `remove_unused_original_minimum_age_seconds` option. 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. For more information about the VMware VMDK driver, see 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 3.1. To 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-messagetypes.
xsd
wget  --no-check-certificate https://$VMWAREAPI_IP/sdk/query-types.xsd
wget  --no-check-certificate https://$VMWAREAPI_IP/sdk/vim-messagetypes.
xsd
wget  --no-check-certificate https://$VMWAREAPI_IP/sdk/vim-types.xsd
wget  --no-check-certificate https://$VMWAREAPI_IP/sdk/reflect-
messagetypes.xsd
wget  --no-check-certificate https://$VMWAREAPI_IP/sdk/reflect-types.xsd
```

   Because the `reflect-types.xsd` and `reflect-messagetypes.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>
```

6. 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
```

Configuration reference

To customize the VMware driver, use the configuration option settings documented in Table 3.58, “Description of VMware configuration options” [312].

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 2008 R2**

  Both Server and Server Core with the Hyper-V role enabled (Shared Nothing Live migration is not supported using 2008 R2)

- **Windows Server 2012 and Windows Server 2012 R2**

  Server and Core (with the Hyper-V role enabled), and Hyper-V Server

Hyper-V configuration

The only OpenStack services required on a Hyper-V node are `nova-compute` and `neutron-hyperv-agent`. Regarding the resources needed for this host you have to consider that Hyper-V will require 16 GB - 20 GB of disk space for the OS itself, including updates. Two NICs are required, one connected to the management network and one to the guest data network.

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 the Windows 2008 R2, 2012 and 2012 R2 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 OpenStack nodes. To set network time on your Windows 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
```

Keep in mind that the node will have to be time synchronized with the other nodes of your OpenStack environment, so it is important to use the same NTP server. Note that in case of an Active Directory environment, you may do this only for the AD Domain Controller.

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 YOUR_BRIDGE_NAME -
AllowManagementOS $false
```

Note

It is very important to make sure that when you are using an Hyper-V node with only 1 NIC the -AllowManagementOS option is set on `True`, otherwise you will lose connectivity to the Hyper-V node.

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.

```
PS C:\>Set-Service -Name MSiSCSI -StartupType Automatic
PS C:\>Start-Service MSiSCSI
```

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 R2 or Windows Server 2012 R2 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 live migration.

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

Install nova-compute using OpenStack Hyper-V installer

In case you want to avoid all the manual setup, you can use Cloudbase Solutions' installer. You can find it here:

https://www.cloudbase.it/downloads/HyperVNovaCompute_Beta.msi

It installs an independent Python environment, in order to avoid conflicts with existing applications, generates dynamically a `nova.conf` file based on the parameters provided by you.

The installer can also be used for an automated and unattended mode for deployments on a massive number of servers. More details about how to use the installer and its features can be found here:

https://www.cloudbase.it

Requirements

Python

Python 2.7 32bit must be installed as most of the libraries are not working properly on the 64bit version.

Procedure 3.2. Setting up Python prerequisites

1. Download and then install it using the MSI installer from here:

   http://www.python.org/ftp/python/2.7.3/python-2.7.3.msi

```
PS C:\> $src = "http://www.python.org/ftp/python/2.7.3/python-2.7.3.msi"
PS C:\> $dest = "$env:temp\python-2.7.3.msi"
PS C:\> Invoke-WebRequest –Uri $src –OutFile $dest
PS C:\> Unblock-File $dest
PS C:\> Start-Process $dest
```

2. Make sure that the `Python` and `Python\Scripts` paths are set up in the PATH environment variable.

```
PS C:\>$oldPath = [System.Environment]::GetEnvironmentVariable("Path")
PS C:\>$newPath = $oldPath + ";C:\python27\;C:\python27\Scripts\"
PS C:\>[System.Environment]::SetEnvironmentVariable("Path", $newPath,
 [System.EnvironmentVariableTarget]::User
```

Python dependencies

The following packages need to be downloaded and manually installed:

setuptools http://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11.win32-py2.7.exel

pip http://pip.readthedocs.org/en/latest/installing.html

MySQL-python http://codegood.com/download/10/

PyWin32 http://sourceforge.net/projects/pywin32/files/pywin32/Build%20217/pywin32-217.win32-py2.7.exe

Greenlet http://www.lfd.uci.edu/~gohlke/pythonlibs/#greenlet

PyCryto http://www.voidspace.org.uk/downloads/pycrypto26/pycrypto-2.6.win32-py2.7.exe

The following packages must be installed with pip:

- ecdsa
- amqp
- wmi

```
PS C:\> pip install ecdsa
PS C:\> pip install amqp
PS C:\> pip install wmi
```

Other dependencies

qemu-img is required for some of the image related operations. You can get it from here: http://qemu.weilnetz.de/. You must make sure that the qemu-img path is set in the PATH environment variable.

Some Python packages need to be compiled, so you may use MinGW or Visual Studio. You can get MinGW from here: http://sourceforge.net/projects/mingw/. You must configure which compiler to be used for this purpose by using the distutils.cfg file in $Python27\Lib\distutils, which can contain:

```
[build]
compiler = mingw32
```

As a last step for setting up MinGW, make sure that the MinGW binaries' directories are set up in PATH.

Install Nova-compute

Download the nova code

1. Use Git to download the necessary source code. The installer to run Git on Windows can be downloaded here:

   https://github.com/msysgit/msysgit/releases/download/Git-1.9.2-preview20140411/Git-1.9.2-preview20140411.exe

2. Download the installer. Once the download is complete, run the installer and follow the prompts in the installation wizard. The default should be acceptable for the needs of the document.

   ```
   PS C:\>$src = "https://github.com/msysgit/msysgit/releases/download/Git-1.
   9.2-preview20140411/Git-1.9.2-preview20140411.exe"
   PS C:\>$dest = "$env:temp\Git-1.9.2-preview20140411.exe"
   PS C:\>Invoke-WebRequest –Uri $src –OutFile $dest
   PS C:\>Unblock-File $dest
   PS C:\>Start-Process $dest
   ```

3. Run the following to clone the Nova code.

   ```
   PS C:\>git.exe clone https://github.com/openstack/nova.git
   ```

Install nova-compute service

To install Nova-compute, run:

```
PS C:\>cd c:\Nova
PS C:\>python setup.py install
```

Configure nova-compute

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]
[DEFAULT]
auth_strategy = keystone
image_service = nova.image.glance.GlanceImageService
```

```
compute_driver = nova.virt.hyperv.driver.HyperVDriver
volume_api_class = nova.volume.cinder.API
fake_network = true
instances_path = C:\Program Files (x86)\OpenStack\Instances
glance_api_servers = IP_ADDRESS:9292
use_cow_images = true
force_config_drive = false
injected_network_template = C:\Program Files (x86)\OpenStack\Nova\etc\
interfaces.template
policy_file = C:\Program Files (x86)\OpenStack\Nova\etc\policy.json
mkisofs_cmd = C:\Program Files (x86)\OpenStack\Nova\bin\mkisofs.exe
verbose = false
allow_resize_to_same_host = true
running_deleted_instance_action = reap
running_deleted_instance_poll_interval = 120
resize_confirm_window = 5
resume_guests_state_on_host_boot = true
rpc_response_timeout = 1800
lock_path = C:\Program Files (x86)\OpenStack\Log\
rpc_backend = nova.openstack.common.rpc.impl_kombu
rabbit_host = IP_ADDRESS
rabbit_port = 5672
rabbit_userid = guest
rabbit_password = Passw0rd
logdir = C:\Program Files (x86)\OpenStack\Log\
logfile = nova-compute.log
instance_usage_audit = true
instance_usage_audit_period = hour
network_api_class = nova.network.neutronv2.api.API
[neutron]
url = http://IP_ADDRESS:9696
auth_strategy = keystone
admin_tenant_name = service
admin_username = neutron
admin_password = Passw0rd
admin_auth_url = http://IP_ADDRESS:35357/v2.0
[hyperv]
vswitch_name = newVSwitch0
limit_cpu_features = false
config_drive_inject_password = false
qemu_img_cmd = C:\Program Files (x86)\OpenStack\Nova\bin\qemu-img.exe
config_drive_cdrom = true
dynamic_memory_ratio = 1
enable_instance_metrics_collection = true
[rdp]
enabled = true
html5_proxy_base_url = https://IP_ADDRESS:4430
```

Table 3.31, "Description of HyperV configuration options" [296] contains a reference of all options for hyper-v.

Prepare images for use with Hyper-V

Hyper-V currently supports only the VHD and VHDX 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:

```
PS C:\>glance image-create --name "VM_IMAGE_NAME" --is-public False --
container-format bare --disk-format vhd
```

Note

VHD and VHDX files sizes can be bigger than their maximum internal size, as such you need to boot instances using a flavor with a slightly bigger disk size than the internal size of the disk file. To create VHDs, use the following PowerShell cmdlet:

```
PS C:\>New-VHD DISK_NAME.vhd -SizeBytes VHD_SIZE
```

Run Compute with Hyper-V

To start the `nova-compute` service, run this command from a console in the Windows server:

```
PS C:\>C:\Python27\python.exe c:\Python27\Scripts\nova-compute --config-file
 c:\etc\nova\nova.conf
```

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. For instructions about how to configure NTP on your Hyper-V compute node, see the section called "Configure NTP" [250].

- How do I restart the compute service?

  ```
  PS C:\>net stop nova-compute && net start nova-compute
  ```

- How do I restart the iSCSI initiator service?

  ```
  PS C:\>net stop msiscsi && net start msiscsi
  ```

Scheduling

Compute uses the `nova-scheduler` service to determine how to dispatch compute requests. For example, the `nova-scheduler` service determines on which host a VM should launch. In the context of filters, the term *host* 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_task_period = 60
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`).
- Can service 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`).
- Are on a different host than other instances of a group (if requested) (`ServerGroupAntiAffinityFilter`).
- Are in a set of group hosts (if requested) (`ServerGroupAffinityFilter`).

The scheduler caches its list of available hosts; use the `scheduler_driver_task_period` option to specify how often the list is updated.

Note

Do not configure `service_down_time` to be much smaller than `scheduler_driver_task_period`; otherwise, hosts appear to be dead while the host list is being cached.

For information about the volume scheduler, see the Block Storage section of *OpenStack Cloud Administrator Guide*.

The scheduler chooses a new host when an instance is migrated.

When evacuating instances from a host, the scheduler service honors the target host defined by the administrator on the evacuate command. If a target is not defined by the administrator, the scheduler determines the target host. For information about instance evacuation, see Evacuate instances section of the *OpenStack Cloud Administrator Guide*.

Filter scheduler

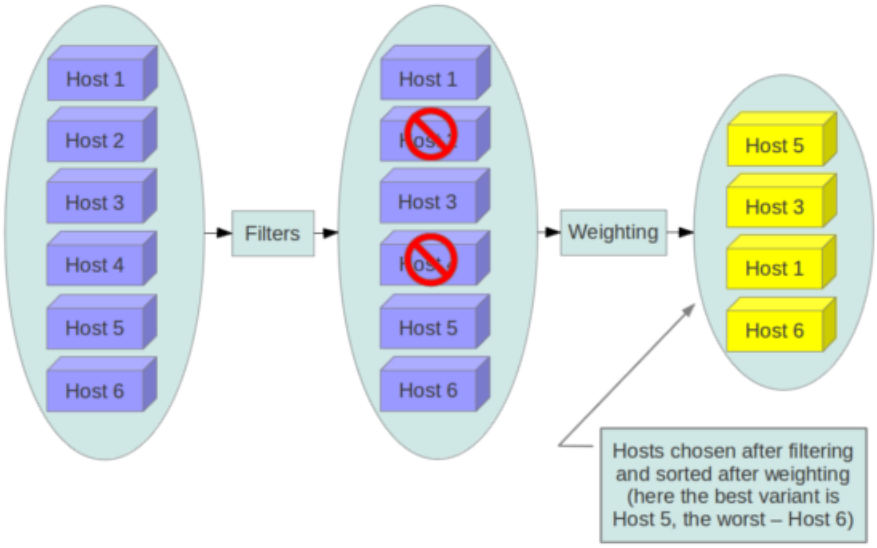
The filter scheduler (`nova.scheduler.filter_scheduler.FilterScheduler`) is the default scheduler for scheduling virtual machine instances. It supports filtering and weighting to make informed decisions on where a new instance should be created.

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 3.2. 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

Filters host by CPU core numbers with a per-aggregate `cpu_allocation_ratio` value. If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and more than one value is found, the minimum value will be used. For information about how to use this filter, see the section called “Host aggregates and availability zones” [271]. See also the section called “CoreFilter” [261].

AggregateDiskFilter

Filters host by disk allocation with a per-aggregate `disk_allocation_ratio` value. If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and more than one value is found, the minimum value will be used. For information about how to use this filter, see the section called "Host aggregates and availability zones" [271]. See also the section called "DiskFilter" [262].

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 matches 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 boots 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 by 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 extra specs for an instance type against admin-defined properties on a host aggregate. Works with specifications that are scoped with `aggregate_instance_extra_specs`. For backward compatibility, also works with non-

scoped specifications; this action is highly discouraged because it conflicts with ComputeCapabilitiesFilter filter when you enable both filters. For information about how to use this filter, see the host aggregates section.

AggregateIoOpsFilter

Filters host by disk allocation with a per-aggregate `max_io_ops_per_host` value. If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and more than one value is found, the minimum value will be used. For information about how to use this filter, see the section called "Host aggregates and availability zones" [271]. See also the section called "IoOpsFilter" [265].

AggregateMultiTenancyIsolation

Isolates tenants to specific host aggregates. If a host is in an aggregate that has the `filter_tenant_id` metadata key, the host creates instances from only 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, the host can create instances from all tenants.

AggregateNumInstancesFilter

Filters host by number of instances with a per-aggregate `max_instances_per_host` value. If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and thus more than one value is found, the minimum value will be used. For information about how to use this filter, see the section called "Host aggregates and availability zones" [271]. See also the section called "NumInstancesFilter" [266].

AggregateRamFilter

Filters host by RAM allocation of instances with a per-aggregate `ram_allocation_ratio` value. If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and thus more than one value is found, the minimum value will be used. For information about how to use this filter, see the section called "Host aggregates and availability zones" [271]. See also the section called "RamFilter" [266].

AggregateTypeAffinityFilter

Filters host by per-aggregate `instance_type` value. For information about how to use this filter, see the section called "Host aggregates and availability zones" [271]. See also the section called "TypeAffinityFilter" [268].

AllHostsFilter

This is a no-op filter. It does not eliminate any of the available hosts.

AvailabilityZoneFilter

Filters hosts by availability zone. You must enable this filter for the scheduler to respect availability zones in requests.

ComputeCapabilitiesFilter

Matches properties defined in extra specs for an instance type 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`, the filter ignores the namespace. For backward compatibility, also treats the extra specs key as the key to be matched if no namespace is present; this action is highly discouraged because it conflicts with AggregateInstanceExtraSpecsFilter filter when you enable both filters.

ComputeFilter

Passes all hosts that are operational and enabled.

In general, you should always enable this filter.

CoreFilter

Only schedules instances on hosts if sufficient CPU cores are available. If this filter is not set, the scheduler might over-provision a host based on cores. For example, the virtual cores running on an instance may exceed the physical cores.

You can configure this filter to enable 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 always returns 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.

NUMATopologyFilter

Filters hosts based on the NUMA topology that was specified for the instance through the use of flavor `extra_specs`in combination with the image properties, as described in detail in the related nova-spec document: Filter will try to match the exact NUMA cells of the instance to those of the host. It will consider the standard over-subscription limits each cell, and provide limits to the compute host accordingly.

Note

If instance has no topology defined, it will be considered for any host. If instance has a topology defined, it will be considered only for NUMA capable hosts.

DifferentHostFilter

Schedules 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 schedules instances on hosts if there is sufficient disk space available for root and ephemeral storage.

You can configure this filter to enable a fixed amount of disk overcommitment by using the `disk_allocation_ratio` configuration option in the `nova.conf` configuration file. The default setting disables the possibility of the overcommitment and allows launching a VM only if there is a sufficient amount of disk space available on a host:

```
disk_allocation_ratio = 1.0
```

DiskFilter always considers the value of the `disk_available_least` property and not the one of the `free_disk_gb` property of a hypervisor's statistics:

```
$ nova hypervisor-stats
+----------------------+-------+
| Property             | Value |
+----------------------+-------+
| count                |  1    |
| current_workload     |  0    |
| disk_available_least |  29   |
```

```
| free_disk_gb          |  35   |
| free_ram_mb           |  3441 |
| local_gb              |  35   |
| local_gb_used         |  0    |
| memory_mb             |  3953 |
| memory_mb_used        |  512  |
| running_vms           |  0    |
| vcpus                 |  2    |
| vcpus_used            |  0    |
+-----------------------+-------+
```

As it can be viewed from the command output above, the amount of the available disk space can be less than the amount of the free disk space. It happens because the `disk_available_least` property accounts for the virtual size rather than the actual size of images. If you use an image format that is sparse or copy on write so that each virtual instance does not require a 1:1 allocation of a virtual disk to a physical storage, it may be useful to allow the overcommitment of disk space.

To enable scheduling instances while overcommitting disk resources on the node, adjust the value of the `disk_allocation_ratio` configuration option to greater than `1.0`:

```
disk_allocation_ratio > 1.0
```

Note

If the value is set to >1, we recommend keeping track of the free disk space, as the value approaching `0` may result in the incorrect functioning of instances using it at the moment.

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, hypervisor version (for Xen hypervisor type only), and virtual machine mode.

For example, an instance might require a host that runs an ARM-based processor, and QEMU as the hypervisor. You can decorate an image 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`: describes the machine architecture required by the image. Examples are `i686`, `x86_64`, `arm`, and `ppc64`.
- `hypervisor_type`: describes the hypervisor required by the image. Examples are `xen`, `qemu`, and `xenapi`.

  Note

  `qemu` is used for both QEMU and KVM hypervisor types.

- `hypervisor_version_requires`: describes the hypervisor version required by the image. The property is supported for Xen hypervisor type only. It can be used to enable support for multiple hypervisor versions, and to prevent instances with newer Xen tools from being provisioned on an older version of a hypervisor. If available, the property value is compared to the hypervisor version of the compute host.

  To filter the hosts by the hypervisor version, add the `hypervisor_version_requires` property on the image as metadata and pass an operator and a required hypervisor version as its value:

  ```
  $ glance image-update img-uuid --property hypervisor_type=xen --property
   hypervisor_version_requires=">=4.3"
  ```

- `vm_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 iso-

lated 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
```

IoOpsFilter

The IoOpsFilter filters hosts by concurrent I/O operations on it. Hosts with too many concurrent I/O operations will be filtered out. The `max_io_ops_per_host` option specifies the maximum number of I/O intensive instances allowed to run on a host. A host will be ignored by the scheduler if more than `max_io_ops_per_host` instances in build, resize, snapshot, migrate, rescue or unshelve task states are running on it.

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]"
    }
}
```

MetricsFilter

Filters hosts based on meters `weight_setting`. Only hosts with the available meters are passed so that the metrics weigher will not fail due to these hosts.

NumInstancesFilter

Hosts that have more instances running than specified by the `max_instances_per_host` option are filtered out when this filter is in place.

PciPassthroughFilter

The filter schedules instances on a host if the host has devices that meet the device requests in the `extra_specs` attribute for the flavor.

RamFilter

Only schedules 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).

You can configure this filter to enable 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

Filters 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

Schedules 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

Schedules 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"
    }
}
```

TrustedFilter

Filters hosts based on their trust. Only passes hosts that meet the trust requirements specified in the instance properties.

TypeAffinityFilter

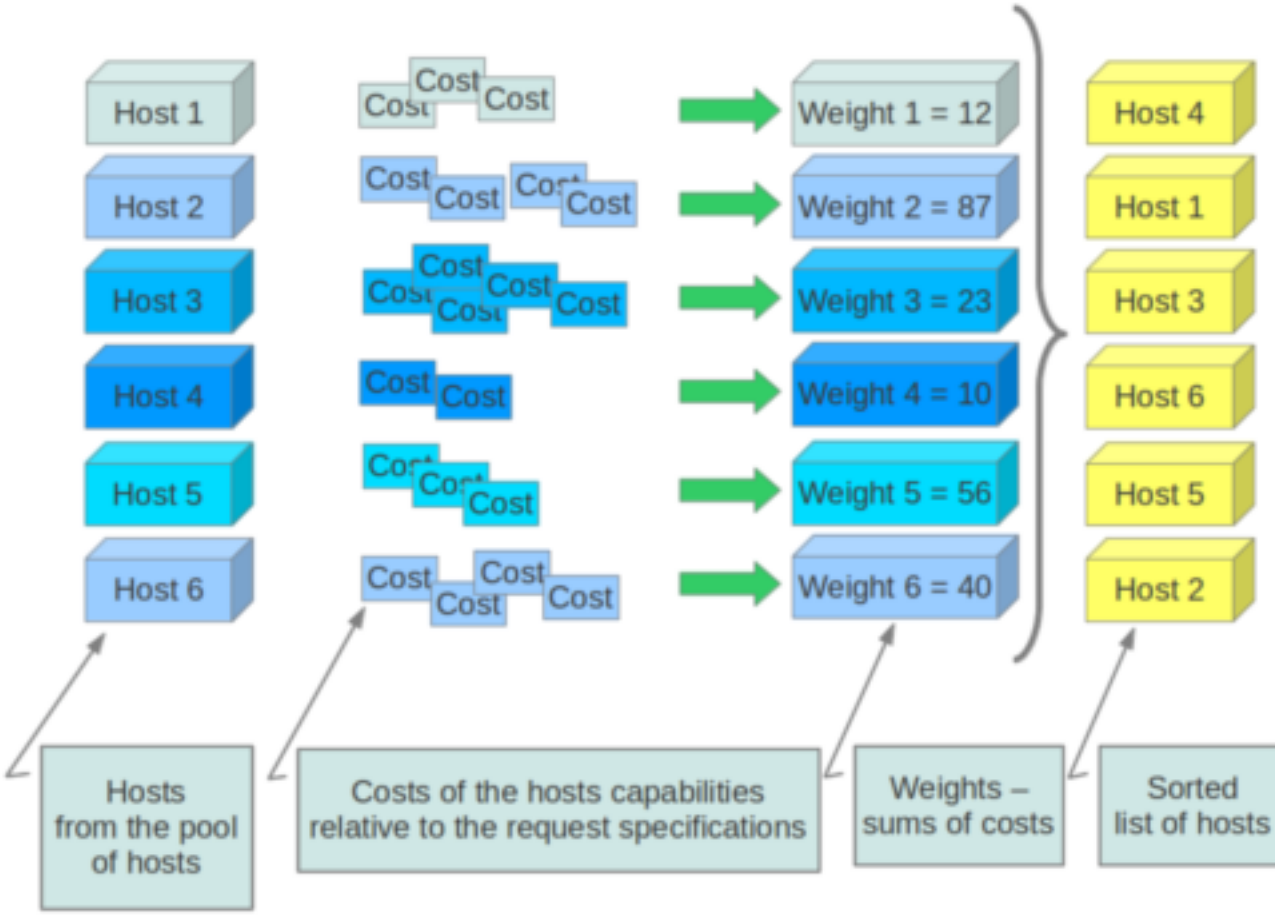
Dynamically limits hosts to one instance type. An instance can only be launched on a host, if no instance with different instances types are running on it, or if the host has no running instances at all.

Weights

When resourcing instances, the filter scheduler filters and weights 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 3.3. Weighting hosts



If cells are used, cells are weighted by the scheduler in the same manner as hosts.

Hosts and cells are weighted based on the following options in the `/etc/nova/nova.conf` file:

Table 3.9. Host weighting options

Section	Option	Description
[DEFAULT]	`ram_weight_multiplier`	By default, the scheduler spreads instances across all hosts evenly. Set the `ram_weight_multiplier` option to a negative number if you prefer stacking instead of spreading. Use a floating-point value.
[DEFAULT]	`scheduler_host_subset_size`	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 weighting 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]	`scheduler_weight_classes`	Defaults to `nova.scheduler.weights.all_weighers`, which selects the RamWeigher and MetricsWeigher. Hosts are then weighted and sorted with the largest weight winning.
[metrics]	`weight_multiplier`	Multiplier for weighting meters. Use a floating-point value.

Section	Option	Description
[metrics]	`weight_setting`	Determines how meters are weighted. Use a comma-separated list of metricName=ratio. 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 meters: • True—Raises an exception. To avoid the raised exception, you should use the scheduler filter `MetricFilter` to filter out hosts with unavailable meters. • False—Treated as a negative factor in the weighting process (uses the `weight_of_unavailable` option).
[metrics]	`weight_of_unavailable`	If `required` is set to False, and any one of the meters set by `weight_setting` is unavailable, the `weight_of_unavailable` 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 3.10. Cell weighting options

Section	Option	Description
[cells]	`mute_weight_multiplier`	Multiplier to weight mute children (hosts which have not sent capacity or capacity updates for some time). Use a negative, floating-point value.
[cells]	`mute_weight_value` (deprecated)	Weight value assigned to mute children. Use a positive, floating-point value with a maximum of '1.0'. This option is deprecated, use `mute_weight_multiplier` instead.
[cells]	`offset_weight_multiplier`	Multiplier to weight 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 `ram_weight_multiplier` option to a negative number if you prefer stacking instead of spreading. Use a floating-point value.
[cells]	`scheduler_weight_classes`	Defaults to `nova.cells.weights.all_weighers`, which maps to all cell weighers included with Compute. Cells are then weighted and sorted with the largest weight winning.

For example:

```
[cells]
scheduler_weight_classes = nova.cells.weights.all_weighers
mute_weight_multiplier = -10.0
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 and availability zones

Host aggregates are a mechanism for partitioning hosts in an OpenStack cloud, or a region of an OpenStack cloud, based on arbitrary characteristics. Examples where an administrator may want to do this include where a group of hosts have additional hardware or performance characteristics.

Host aggregates are not explicitly exposed to users. Instead administrators map flavors to host aggregates. Administrators do this by setting metadata on a host aggregate, and matching flavor extra specifications. The scheduler then endeavors to match user requests for instance of the given flavor to a host aggregate with the same key-value pair in its metadata. Compute nodes can be in more than one host aggregate.

Administrators are able to optionally expose a host aggregate as an availability zone. Availability zones are different from host aggregates in that they are explicitly exposed to the user, and hosts can only be in a single availability zone. Administrators can configure a default availability zone where instances will be scheduled when the user fails to specify one.

Command-line interface

The **nova** command-line tool supports the following aggregate-related commands.

nova aggregate-list	Print a list of all aggregates.
nova aggregate-create <*name*> [*availability-zone*]	Create a new aggregate named <*name*>, and optionally in availability zone *[availability-zone]* if specified. The command returns the ID of the newly created aggregate. Hosts can be made available to multiple host aggregates. Be careful when adding a host to an additional host aggregate when the host is also in an availability zone. Pay attention when using the **aggregate-set-metadata** and **aggregate-update** 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.
nova aggregate-delete <*id*>	Delete an aggregate with id <*id*>.
nova aggregate-details <*id*>	Show details of the aggregate with id <*id*>.
nova aggregate-add-host <*id*> <*host*>	Add host with name <*host*> to aggregate with id <*id*>.
nova aggregate-remove-host <*id*> <*host*>	Remove the host with name <*host*> from the aggregate with id <*id*>.
nova aggregate-set-metadata <*id*> <*key=value*> [<*key=value*> ...]	Add or update metadata (key-value pairs) associated with the aggregate with id <*id*>.
nova aggregate-update <*id*> <*name*> [<*availability_zone*>]	Update the name and availability zone (optional) for the aggregate.

nova host-list	List all hosts by service.
nova host-update –maintenance [enable | disable]	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,RetryFilter,
AvailabilityZoneFilter,RamFilter,ComputeFilter,ComputeCapabilitiesFilter,
ImagePropertiesFilter,ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter
```

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              | []    | {u'ssd': u'true'} |
```

```
+----+---------+------------------+-------+------------------+

$ nova aggregate-add-host 1 node1
+----+---------+-------------------+-----------+------------------+
| Id | Name    | Availability Zone | Hosts     | Metadata         |
+----+---------+-------------------+-----------+------------------+
| 1  | fast-io | nova              | [u'node1'] | {u'ssd': u'true'} |
+----+---------+-------------------+-----------+------------------+

$ nova aggregate-add-host 1 node2
+----+---------+-------------------+---------------------+------------------+
| Id | Name    | Availability Zone | Hosts               | Metadata
 |
+----+---------+-------------------+---------------------+------------------
+
| 1  | fast-io | nova              | [u'node1', u'node2'] | {u'ssd': u'true'}
 |
+----+---------+-------------------+---------------------+------------------
+
```

Use the **nova flavor-create** command to create the *ssd.large* flavor called with an ID of 6, 8 GB of RAM, 80 GB root disk, and four vCPUs.

```
$ nova flavor-create ssd.large 6 8192 80 4
+----+-----------+-----------+------+-----------+------+-------+-------------
+-----------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor |
 Is_Public |
+----+-----------+-----------+------+-----------+------+-------+-------------
+-----------+
| 6  | ssd.large | 8192      | 80   | 0         |      | 4     | 1.0         |
 True      |
+----+-----------+-----------+------+-----------+------+-------+-------------
+-----------+
```

Once the flavor is created, specify one or more key-value pairs that match the key-value pairs on the host aggregates with scope aggregate_instance_extra_specs. In this case, that is the *aggregate_instance_extra_specs:ssd=true* key-value pair. Setting a key-value pair on a flavor is done using the **nova flavor-key** command.

```
$ nova flavor-key ssd.large set  aggregate_instance_extra_specs:ssd=true
```

Once it is set, you should see the `extra_specs` property of the `ssd.large` flavor populated with a key of `ssd` and a corresponding value of `true`.

```
$ nova flavor-show ssd.large
+----------------------------
+-------------------------------------------------+
| Property                   | Value
  |
+----------------------------
+-------------------------------------------------+
| OS-FLV-DISABLED:disabled   | False
  |
| OS-FLV-EXT-DATA:ephemeral  | 0
  |
| disk                       | 80
  |
| extra_specs                | {u'aggregate_instance_extra_specs:ssd':
 u'true'} |
```

```
| id                          | 6
  |
| name                        | ssd.large
  |
| os-flavor-access:is_public | True
  |
| ram                         | 8192
  |
| rxtx_factor                 | 1.0
  |
| swap                        |
  |
| vcpus                       | 4
  |
+-----------------------------
+-------------------------------------------------+
```

Now, when a user requests an instance with the `ssd.large` flavor, the scheduler only considers hosts with the `ssd=true` key-value pair. In this example, these are `node1` and `node2`.

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 3.52, "Description of scheduler configuration options" [309].

Cells

Cells functionality enables 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 supports 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 appear in the `[cells]` section in `nova.conf`. The following cell-related options are currently supported:

enable	Set to `True` to turn on cell functionality. Default is `false`.
name	Name of the current cell. Must be unique for each cell.
capabilities	List of arbitrary *key=value* pairs defining capabilities of the current cell. Values include `hypervisor=xenserver;kvm,os=linux;windows`.
call_timeout	How long in seconds to wait for replies from calls between cells.
scheduler_filter_classes	Filter classes that the cells scheduler should use. By default, uses "`nova.cells.filters.all_filters`" to map to all cells filters included with Compute.
scheduler_weight_classes	Weight classes that the scheduler for cells uses. By default, uses `nova.cells.weights.all_weighers` to map to all cells weight algorithms included with Compute.
ram_weight_multiplier	Multiplier used to weight RAM. Negative numbers indicate that Compute should stack VMs on one host instead of spreading out new VMs to more hosts in the cell. The default value is 10.0.

Configure the API (top-level) cell

The cell type must be changed in the API cell so that requests can be proxied through nova-cells down to the correct cell properly. Edit the `nova.conf` file in the API cell, and specify `api` in the `cell_type` key:

```
[DEFAULT]
compute_api_class=nova.compute.cells_api.ComputeCellsAPI
...

[cells]
cell_type= api
```

Configure the child cells

Edit the `nova.conf` file in the child cells, and specify `compute` in the `cell_type` key:

```
[DEFAULT]
# Disable quota checking in child cells. Let API cell do it exclusively.
quota_driver=nova.quota.NoopQuotaDriver

[cells]
cell_type = compute
```

Configure the database in each cell

Before bringing the services online, the database in each cell needs to be configured with information about related cells. In particular, the API cell needs to know about its immediate children, and the child cells must know about their immediate agents. The information needed is the RabbitMQ server credentials for the particular cell.

Use the **nova-manage cell create** command to add this information to the database in each cell:

```
# nova-manage cell create -h
Options:
  -h, --help            show this help message and exit
  --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 an API cell named `api` and a child cell named `cell1`.

Within the `api` cell, specify the following RabbitMQ server information:

```
rabbit_host=10.0.0.10
rabbit_port=5672
rabbit_username=api_user
rabbit_password=api_passwd
rabbit_virtual_host=api_vhost
```

Within the `cell1` child cell, specify the following RabbitMQ server information:

```
rabbit_host=10.0.1.10
rabbit_port=5673
rabbit_username=cell1_user
rabbit_password=cell1_passwd
rabbit_virtual_host=cell1_vhost
```

You can 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 previous steps for all child cells.

In the child cell, 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 3.19, "Description of cell configuration options" [289].

Cell scheduling configuration

To determine the best cell to use to launch a new instance, Compute uses a set of filters and weights defined in the `/etc/nova/nova.conf` file. The following options are available to prioritize cells for scheduling:

`scheduler_filter_classes` List of filter classes. By default `nova.cells.filters.all_filters` is specified, which maps to all cells filters included with Compute (see the section called "Filters" [257]).

`scheduler_weight_classes` List of weight classes. By default `nova.cells.weights.all_weighers` is specified, which maps to all cell weight algorithms 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).

  Warning

  The `mute_weight_value` is deprecated, use `mute_weight_multiplier` instead.

- `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, '999999999999999'). 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 store all inter-cell communication data, including user names and passwords, in the database. Because the cells data is not updated very frequently, use the `[cells]cells_config` option to specify a JSON file to store cells data. With this configuration, the database is no longer consulted when reloading the cells data. The file must have columns present in the Cell model (excluding common database fields and the `id` column). You must specify the queue connection information 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 3.22, "Description of conductor configuration options" [292].

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

# 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

# GLANCE
[glance]
api_servers=192.168.206.130:9292

# 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

# RABBITMQ
rabbit_host=192.168.206.130

# GLANCE
image_service=nova.image.glance.GlanceImageService

# 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
```

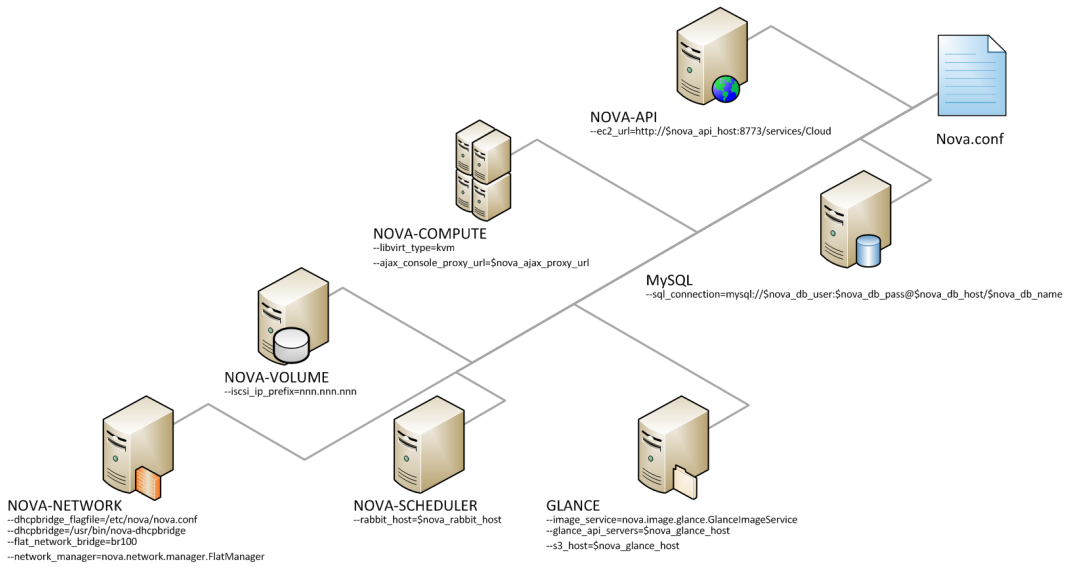
```
# AUTHENTICATION
auth_strategy=keystone
[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = nova
signing_dirname = /tmp/keystone-signing-nova

# GLANCE
[glance]
api_servers=192.168.206.130:9292

# DATABASE
[database]
connection=mysql://nova:yourpassword@192.168.206.130/nova

# LIBVIRT
[libvirt]
virt_type=qemu
```

Figure 3.4. KVM, Flat, MySQL, and Glance, OpenStack or EC2 API



XenServer, Flat networking, MySQL, and Glance, OpenStack API

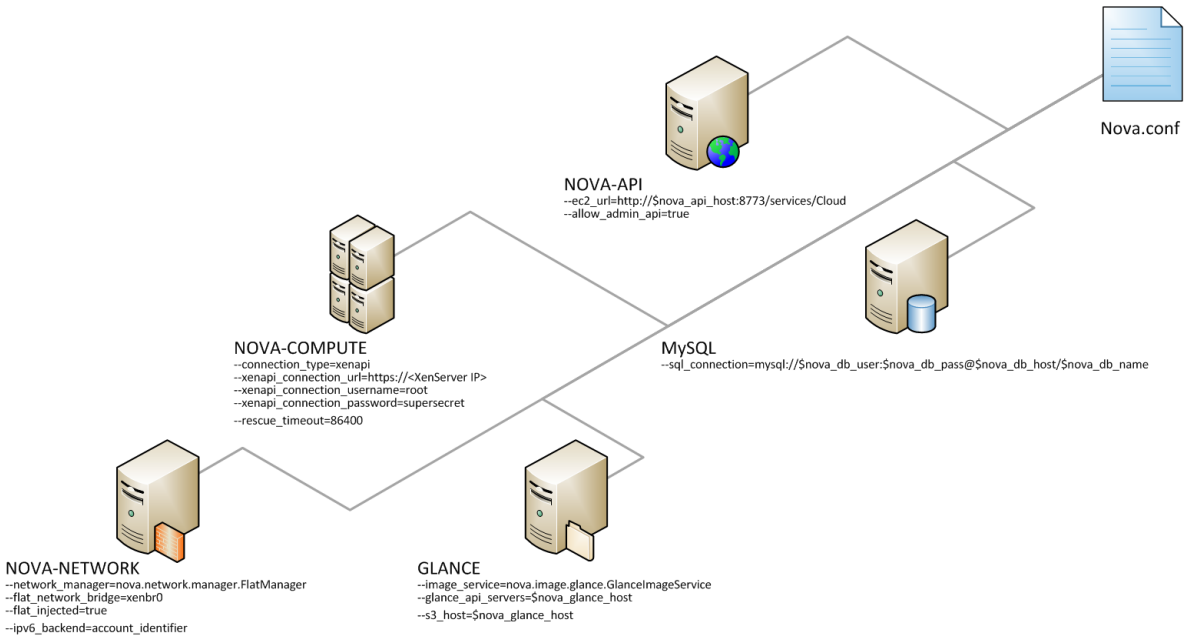
This example `nova.conf` file is from an internal Rackspace test system.

```
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 3.5. 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 3.11. Log files used by Compute services

Log file	Service name (CentOS/Fedora/openSUSE/Red Hat Enterprise Linux/SUSE Linux Enterprise)	Service name (Ubuntu/Debian)
`api.log`	openstack-nova-api	nova-api
`cert.log` [a]	`openstack-nova-cert`	`nova-cert`
`compute.log`	`openstack-nova-compute`	`nova-compute`
`conductor.log`	`openstack-nova-conductor`	`nova-conductor`
`consoleauth.log`	`openstack-nova-consoleauth`	`nova-consoleauth`
`network.log`[b]	`openstack-nova-network`	`nova-network`
`nova-manage.log`	`nova-manage`	`nova-manage`
`scheduler.log`	`openstack-nova-scheduler`	`nova-scheduler`

[a]The X509 certificate service (`openstack-nova-cert`/`nova-cert`) is only required by the EC2 API to the Compute service.

[b]The `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 3.12. Description of API configuration options

Configuration option = Default value	Description
[DEFAULT]	
`api_paste_config` = *`api-paste.ini`*	(StrOpt) File name for the paste.deploy config for nova-api
`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.
`client_socket_timeout` = *`900`*	(IntOpt) Timeout for client connections' socket operations. If an incoming connection is idle for this number of seconds it will be closed. A value of '0' means wait forever.
`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).

Configuration option = Default value	Description
multi_instance_display_name_template = *%(name)s-%(count)d*	(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)
snapshot_name_template = *snapshot-%s*	(StrOpt) Template string to be used to generate snapshot names
tcp_keepidle = *600*	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
use_forwarded_for = *False*	(BoolOpt) Treat X-Forwarded-For as the canonical remote address. Only enable this if you have a sanitizing proxy.
wsgi_default_pool_size = *1000*	(IntOpt) Size of the pool of greenthreads used by wsgi
wsgi_keep_alive = *True*	(BoolOpt) If False, closes the client socket connection explicitly.
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 3.13. Description of API v3 configuration options

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 3.14. Description of authentication configuration options

Configuration option = Default value	Description
[DEFAULT]	
auth_strategy = *keystone*	(StrOpt) The strategy to use for auth: keystone, noauth (deprecated), or noauth2. Both noauth and noauth2 are designed for testing only, as they do no actual credential checking. noauth provides administrative credentials regardless of the passed in user, noauth2 only does if 'admin' is specified as the username.

Table 3.15. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.

Configuration option = Default value	Description
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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

Configuration option = Default value	Description
	for a configurable duration (in seconds). Set to -1 to disable caching completely.

Table 3.16. Description of availability zones configuration options

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 3.17. Description of Barbican configuration options

Configuration option = Default value	Description
[barbican]	
cafile = *None*	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
catalog_info = *key-manager:barbican:public*	(StrOpt) Info to match when looking for barbican in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type>
certfile = *None*	(StrOpt) PEM encoded client certificate cert file
endpoint_template = *None*	(StrOpt) Override service catalog lookup with template for barbican endpoint e.g. http://localhost:9311/v1/%(project_id)s
insecure = *False*	(BoolOpt) Verify HTTPS connections.
keyfile = *None*	(StrOpt) PEM encoded client certificate key file
os_region_name = *None*	(StrOpt) Region name of this node
timeout = *None*	(IntOpt) Timeout value for http requests

Table 3.18. Description of CA and SSL configuration options

Configuration option = Default value	Description
[DEFAULT]	
ca_file = *cacert.pem*	(StrOpt) Filename of root CA
ca_path = *$state_path/CA*	(StrOpt) Where we keep our root CA
cert = *self.pem*	(StrOpt) SSL certificate file
cert_manager = *nova.cert.manager.CertManager*	(StrOpt) Full class name for the Manager for cert
cert_topic = *cert*	(StrOpt) The topic cert nodes listen on
crl_file = *crl.pem*	(StrOpt) Filename of root Certificate Revocation List
key_file = *private/cakey.pem*	(StrOpt) Filename of private key
keys_path = *$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
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
use_project_ca = *False*	(BoolOpt) Should we use a CA for each project?

Configuration option = Default value	Description
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 3.19. Description of cell configuration options

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.
db_check_interval = *60*	(IntOpt) Interval, in seconds, for getting fresh cell information 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
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 3.20. Description of common configuration options

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

Configuration option = Default value	Description
`console_topic = console`	(StrOpt) The topic console proxy nodes listen on
`consoleauth_topic = consoleauth`	(StrOpt) The topic console auth proxy nodes listen on
`host = localhost`	(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
`memcached_servers = None`	(ListOpt) Memcached servers or None for in process cache.
`my_ip = 10.0.0.1`	(StrOpt) IP address of this host
`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.
`pybasedir = /usr/lib/python/site-packages/nova`	(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
[keystone_authtoken]	
`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.
[workarounds]	
`destroy_after_evacuate = True`	(BoolOpt) Whether to destroy instances on startup when we suspect they have previously been evacuated. This can result in data loss if undesired. See https://launchpad.net/bugs/1419785
`disable_libvirt_livesnapshot = True`	(BoolOpt) When using libvirt 1.2.2 fails live snapshots intermittently under load. This config option provides mechanism to disable livesnapshot while this is resolved. See https://bugs.launchpad.net/nova/+bug/1334398
`disable_rootwrap = False`	(BoolOpt) This option allows a fallback to sudo for performance reasons. For example see https://bugs.launchpad.net/nova/+bug/1415106

Table 3.21. Description of Compute configuration options

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.
`compute_driver = None`	(StrOpt) Driver to use for controlling virtualization. Options include: libvirt.LibvirtDriver, xenapi.XenAPIDriver, fake.FakeDriver, baremetal.BareMetalDriver, vmwareapi.VMwareVCDriver, hyperv.HyperVDriver
`compute_manager = nova.compute.manager.ComputeManager`	(StrOpt) Full class name for the Manager for compute

Configuration option = Default value	Description
compute_monitors =	(ListOpt) A list of monitors that can be used for getting compute metrics.
compute_resources = *vcpu*	(ListOpt) The names of the extra resources to track.
compute_stats_class = *nova.compute.stats.Stats*	(StrOpt) Class that will manage stats for the local compute host
console_host = *localhost*	(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
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 network information cache updates
image_cache_manager_interval = *2400*	(IntOpt) Number of seconds to wait between runs of the image cache manager. Set to -1 to disable. Setting this to 0 will run at the default rate.
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_ $my_ip
instance_build_timeout = *0*	(IntOpt) Amount of time in seconds an instance can be in BUILD before going into ERROR status. Set to 0 to disable.
instance_delete_interval = *300*	(IntOpt) Interval in seconds for retrying failed instance file deletes. Set to -1 to disable. Setting this to 0 will run at the default rate.
instance_usage_audit = *False*	(BoolOpt) Generate periodic compute.instance.exists notifications
instance_usage_audit_period = *month*	(StrOpt) Time period to generate instance usages for. Time period must be hour, day, month or year
instances_path = *$state_path/instances*	(StrOpt) Where instances are stored on disk
max_concurrent_builds = *10*	(IntOpt) Maximum number of instance builds to run concurrently
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
rescue_timeout = *0*	(IntOpt) Automatically unrescue an instance after N seconds. Set to 0 to disable.
resize_confirm_window = *0*	(IntOpt) Automatically confirm resizes after N seconds. Set to 0 to disable.
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.

Configuration option = Default value	Description
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. Set to -1 to disable.Setting this to 0 will run at the default rate.
shutdown_timeout = *60*	(IntOpt) Total amount of time to wait in seconds for an instance to perform a clean shutdown.
sync_power_state_interval = *600*	(IntOpt) Interval to sync power states between the database and the hypervisor. Set to -1 to disable. Setting this to 0 will run at the default rate.
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 3.22. Description of conductor configuration options

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 3.23. Description of config drive configuration options

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
force_config_drive = *None*	(StrOpt) Set to "always" to force injection to take place on a config drive. NOTE: The "always" will be deprecated in the Liberty release cycle.
mkisofs_cmd = *genisoimage*	(StrOpt) Name and optionally path of the tool used for ISO image creation
[hyperv]	
config_drive_cdrom = *False*	(BoolOpt) Attaches the Config Drive image as a cdrom drive instead of a disk drive

Configuration option = Default value	Description
config_drive_inject_password = *False*	(BoolOpt) Sets the admin password in the config drive image

Table 3.24. Description of console configuration options

Configuration option = Default value	Description
[DEFAULT]	
console_public_hostname = *localhost*	(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 3.25. Description of database configuration options

Configuration option = Default value	Description
[DEFAULT]	
db_driver = *nova.db*	(StrOpt) The driver to use for database access
[api_database]	
connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the Nova API 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 number of database connection retries during startup. Set to -1 to specify an infinite retry count.
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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = *True*	(BoolOpt) If True, SQLite uses synchronous mode.
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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.

Configuration option = Default value	Description
db_inc_retry_interval = *True*	(BoolOpt) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify 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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.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.
use_tpool = *False*	(BoolOpt) Enable the experimental use of thread pooling for all DB API calls

Table 3.26. Description of logging configuration options

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.
[guestfs]	
debug = *False*	(BoolOpt) Enable guestfs debug

Table 3.27. Description of EC2 configuration options

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 = */*	(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_insecure = *False*	(BoolOpt) Disable SSL certificate verification.
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 3.28. Description of ephemeral storage encryption configuration options

Configuration option = Default value	Description
[ephemeral_storage_encryption]	
cipher = *aes-xts-plain64*	(StrOpt) The cipher and mode to be used to encrypt ephemeral storage. Which ciphers are available ciphers depends on kernel support. See /proc/crypto for the list of available options.
enabled = *False*	(BoolOpt) Whether to encrypt ephemeral storage
key_size = *512*	(IntOpt) The bit length of the encryption key to be used to encrypt ephemeral storage (in XTS mode only half of the bits are used for encryption key)

Table 3.29. Description of fping configuration options

Configuration option = Default value	Description
[DEFAULT]	
fping_path = */usr/sbin/fping*	(StrOpt) Full path to fping.

Table 3.30. Description of glance configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
osapi_glance_link_prefix = *None*	(StrOpt) Base URL that will be presented to users in links to glance resources
[glance]	
allowed_direct_url_schemes =	(ListOpt) A list of url scheme that can be downloaded directly via the direct_url. Currently supported schemes: [file].
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL (https) requests to glance
api_servers = *None*	(ListOpt) A list of the glance api servers available to nova. Prefix with https:// for ssl-based glance api servers. ([hostname|ip]:port)
host = *$my_ip*	(StrOpt) Default glance hostname or IP address
num_retries = *0*	(IntOpt) Number of retries when uploading / downloading an image to / from glance.
port = *9292*	(IntOpt) Default glance port
protocol = *http*	(StrOpt) Default protocol to use when connecting to glance. Set to https for SSL.
[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 3.31. Description of HyperV configuration options

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 by 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
wait_soft_reboot_seconds = *60*	(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 3.32. Description of hypervisor configuration options

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
`preallocate_images` = *none*	(StrOpt) VM image preallocation mode: "none" => no storage provisioning is done up front, "space" => storage is fully allocated at instance start
`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>

Table 3.33. Description of bare metal configuration options

Configuration option = Default value	Description
[ironic]	
`admin_auth_token` = *None*	(StrOpt) Ironic keystone auth token.
`admin_password` = *None*	(StrOpt) Ironic keystone admin password.
`admin_tenant_name` = *None*	(StrOpt) Ironic keystone tenant name.
`admin_url` = *None*	(StrOpt) Keystone public API endpoint.
`admin_username` = *None*	(StrOpt) Ironic keystone admin name
`api_endpoint` = *None*	(StrOpt) URL for Ironic API endpoint.
`api_max_retries` = *60*	(IntOpt) How many retries when a request does conflict.
`api_retry_interval` = *2*	(IntOpt) How often to retry in seconds when a request does conflict
`api_version` = *1*	(IntOpt) Version of Ironic API service endpoint.
`client_log_level` = *None*	(StrOpt) Log level override for ironicclient. Set this in order to override the global "default_log_levels", "verbose", and "debug" settings. DEPRECATED: use standard logging configuration.

Table 3.34. Description of IPv6 configuration options

Configuration option = Default value	Description
[DEFAULT]	
`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 3.35. Description of key manager configuration options

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

Configuration option = Default value	Description
fixed_key = *None*	(StrOpt) Fixed key returned by key manager, specified in hex

Table 3.36. Description of LDAP configuration options

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 3.37. Description of Libvirt configuration options

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]	
block_migration_flag = *VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_LIVE, VIR_MIGRATE_TUNNELLED, VIR_MIGRATE_NON_SHARED_INC*	(StrOpt) Migration flags to be set for block migration
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"
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_cachemodes =	(ListOpt) Specific cachemodes to use for different disk types e.g: file=directsync,block=none

Configuration option = Default value	Description
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)
gid_maps =	(ListOpt) List of guid targets and ranges.Syntax is guest-gid:host-gid:countMaximum of 5 allowed.
hw_disk_discard = *None*	(StrOpt) Discard option for nova managed disks (valid options are: ignore, unmap). Need Libvirt(1.0.6) Qemu1.5 (raw format) Qemu1.6(qcow2 format)
hw_machine_type = *None*	(ListOpt) For qemu or KVM guests, set this option to specify a default machine type per host architecture. You can find a list of supported machine types in your environment by checking the output of the "virsh capabilities"command. The format of the value for this config option is host-arch=machine-type. For example: x86_64=machinetype1,armv7l=machinetype2
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
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_iface = *None*	(StrOpt) The iSCSI transport iface to use to connect to target in case offload support is desired. Supported transports are be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx and ocs. Default format is transport_name.hwaddress and can be generated manually or via iscsiadm -m iface
iscsi_use_multipath = *False*	(BoolOpt) Use multipath connection of the iSCSI volume
iser_use_multipath = *False*	(BoolOpt) Use multipath connection of the iSER volume
mem_stats_period_seconds = *10*	(IntOpt) A number of seconds to memory usage statistics period. Zero or negative value mean to disable memory usage statistics.
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
rescue_image_id = *None*	(StrOpt) Rescue ami image. This will not be used if an image id is provided by the user.
rescue_kernel_id = *None*	(StrOpt) Rescue aki image
rescue_ramdisk_id = *None*	(StrOpt) Rescue ari image
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

Configuration option = Default value	Description
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
snapshots_directory = *$instances_path/snapshots*	(StrOpt) Location where libvirt driver will store snapshots before uploading them to image service
sparse_logical_volumes = *False*	(BoolOpt) Create sparse logical volumes (with virtualsize) if this flag is set to True.
sysinfo_serial = *auto*	(StrOpt) The data source used to the populate the host "serial" UUID exposed to guest in the virtual BIOS. Permitted options are "hardware", "os", "none" or "auto" (default).
uid_maps =	(ListOpt) List of uid targets and ranges.Syntax is guest-uid:host-uid:countMaximum of 5 allowed.
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
virt_type = *kvm*	(StrOpt) Libvirt domain type (valid options are: kvm, lxc, qemu, uml, xen and parallels)
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
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 3.38. Description of live migration configuration options

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, VIR_MIGRATE_LIVE, VIR_MIGRATE_TUNNELLED*	(StrOpt) Migration flags to be set for live migration
live_migration_uri = *qemu+tcp://%s/system*	(StrOpt) Migration target URI (any included "%s" is replaced with the migration target hostname)

Table 3.39. Description of logging configuration options

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.

Configuration option = Default value	Description
requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
fatal_exception_format_errors = *False*	(BoolOpt) Make exception message format errors fatal
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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) Enables or disables publication of error events.

Configuration option = Default value	Description
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
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 will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 3.40. Description of metadata configuration options

Configuration option = Default value	Description
[DEFAULT]	
metadata_cache_expiration = *15*	(IntOpt) Time in seconds to cache metadata; 0 to disable metadata caching entirely (not recommended). Increasingthis should improve response times of the metadata API when under heavy load. Higher values may increase memoryusage and result in longer times for host metadata changes to take effect.
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 3.41. Description of network configuration options

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

Configuration option = Default value	Description
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 = 86400	(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
ebtables_exec_attempts = 3	(IntOpt) Number of times to retry ebtables commands on failure.
ebtables_retry_interval = 1.0	(FloatOpt) Number of seconds to wait between ebtables retries.
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
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 the 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.

Configuration option = Default value	Description
iptables_top_regex =	(StrOpt) Regular expression to match the 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) DEPRECATED: THIS VALUE SHOULD BE SET WHEN CREATING THE NETWORK. 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
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) DEPRECATED: THIS VALUE SHOULD BE SET WHEN CREATING THE NETWORK. 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]	

Configuration option = Default value	Description
vlan_interface = *vmnic0*	(StrOpt) Physical ethernet adapter name for vlan networking

Table 3.42. Description of neutron configuration options

Configuration option = Default value	Description
[DEFAULT]	
neutron_default_tenant_id = *default*	(StrOpt) Default tenant id when creating neutron networks
[neutron]	
admin_auth_url = *http://localhost:5000/v2.0*	(StrOpt) Authorization URL for connecting to neutron in admin context. DEPRECATED: specify an auth_plugin and appropriate credentials instead.
admin_password = *None*	(StrOpt) Password for connecting to neutron in admin context DEPRECATED: specify an auth_plugin and appropriate credentials instead.
admin_tenant_id = *None*	(StrOpt) Tenant id for connecting to neutron in admin context DEPRECATED: specify an auth_plugin and appropriate credentials instead.
admin_tenant_name = *None*	(StrOpt) Tenant name for connecting to neutron in admin context. This option will be ignored if neutron_admin_tenant_id is set. Note that with Keystone V3 tenant names are only unique within a domain. DEPRECATED: specify an auth_plugin and appropriate credentials instead.
admin_user_id = *None*	(StrOpt) User id for connecting to neutron in admin context. DEPRECATED: specify an auth_plugin and appropriate credentials instead.
admin_username = *None*	(StrOpt) Username for connecting to neutron in admin context DEPRECATED: specify an auth_plugin and appropriate credentials instead.
allow_duplicate_networks = *False*	(BoolOpt) DEPRECATED: Allow an instance to have multiple vNICs attached to the same Neutron network. This option is deprecated in the 2015.1 release and will be removed in the 2015.2 release where the default behavior will be to always allow multiple ports from the same network to be attached to an instance.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
auth_strategy = *keystone*	(StrOpt) Authorization strategy for connecting to neutron in admin context. DEPRECATED: specify an auth_plugin and appropriate credentials instead. If an auth_plugin is specified strategy will be ignored.
cafile = *None*	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = *None*	(StrOpt) PEM encoded client certificate cert file
extension_sync_interval = *600*	(IntOpt) Number of seconds before querying neutron for extensions
insecure = *False*	(BoolOpt) Verify HTTPS connections.
keyfile = *None*	(StrOpt) PEM encoded client certificate key file
metadata_proxy_shared_secret =	(StrOpt) Shared secret to validate proxies Neutron metadata requests
ovs_bridge = *br-int*	(StrOpt) Name of Integration Bridge used by Open vSwitch

Configuration option = Default value	Description
region_name = *None*	(StrOpt) Region name for connecting to neutron in admin context
service_metadata_proxy = *False*	(BoolOpt) Set flag to indicate Neutron will proxy metadata requests and resolve instance ids.
timeout = *None*	(IntOpt) Timeout value for http requests
url = *http://127.0.0.1:9696*	(StrOpt) URL for connecting to neutron

Table 3.43. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
max_request_body_size = *114688*	(IntOpt) The maximum body size for each request, in bytes.

Table 3.44. Description of PCI configuration options

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 3.45. Description of periodic configuration options

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 3.46. Description of policy configuration options

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. This defaults to 0(off) to avoid additional load but it is useful to turn on to help keep quota usage up to date and reduce the impact of out of sync usage issues. Note that quotas are not updated on a periodic task, they

Configuration option = Default value	Description
	will update on a new reservation if max_age has passed since the last reservation
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
password_length = *12*	(IntOpt) Length of generated instance admin passwords
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = *['policy.d']*	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = *policy.json*	(StrOpt) The JSON file that defines policies.
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. This defaults to 0(off) to avoid additional load but it is useful to turn on to help keep quota usage up to date and reduce the impact of out of sync usage issues.

Table 3.47. Description of Quobyte USP volume driver configuration options

Configuration option = Default value	Description
[libvirt]	
quobyte_client_cfg = *None*	(StrOpt) Path to a Quobyte Client configuration file.
quobyte_mount_point_base = *$state_path/mnt*	(StrOpt) Directory where the Quobyte volume is mounted on the compute node

Table 3.48. Description of quota configuration options

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 -1 to disable. Setting this to 0 will run at the default rate.
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

Configuration option = Default value	Description
quota_injected_file_content_bytes = 10240	(IntOpt) Number of bytes allowed per injected file
quota_injected_file_path_length = 255	(IntOpt) Length of 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_networks = 3	(IntOpt) Number of private networks allowed per project
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
quota_server_group_members = 10	(IntOpt) Number of servers per server group
quota_server_groups = 10	(IntOpt) Number of server groups per project
[cells]	
bandwidth_update_interval = 600	(IntOpt) Seconds between bandwidth updates for cells.

Table 3.49. Description of RDP configuration options

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 3.50. Description of Redis configuration options

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 3.51. Description of S3 configuration options

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

Configuration option = Default value	Description
s3_use_ssl = *False*	(BoolOpt) Whether to use SSL when talking to S3

Table 3.52. Description of scheduler configuration options

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
baremetal_scheduler_default_filters = *RetryFilter, AvailabilityZoneFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter, ExactRamFilter, ExactDiskFilter, ExactCoreFilter*	(ListOpt) Which filter class names to use for filtering baremetal hosts when not specified in the request.
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
io_ops_weight_multiplier = *-1.0*	(FloatOpt) Multiplier used for weighing host io ops. Negative numbers mean a preference to choose light workload compute hosts.
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) Tells filters to ignore hosts that have this many or more instances currently in build, resize, snapshot, migrate, rescue or unshelve task states
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
scheduler_available_filters = *['nova.scheduler.filters.all_filters']*	(MultiStrOpt) Filter classes available to the scheduler which may be specified more than once. An entry of "nova.scheduler.filters.all_filters" maps to all filters included with nova.
scheduler_default_filters = *RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter*	(ListOpt) Which filter class names to use for filtering hosts when not specified in the request.
scheduler_driver = *nova.scheduler.filter_scheduler.FilterScheduler*	(StrOpt) Default driver to use for the scheduler
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

Configuration option = Default value	Description
	exactly how they interact will depend on your choice of scheduler driver.
scheduler_host_manager = *nova.scheduler.host_manager.HostManager*	(StrOpt) The scheduler host manager class to use
scheduler_host_subset_size = *1*	(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
scheduler_instance_sync_interval = *120*	(IntOpt) Waiting time interval (seconds) between sending the scheduler a list of current instance UUIDs to verify that its view of instances is in sync with nova. If the CONF option `scheduler_tracks_instance_changes` is False, changing this option will have no effect.
scheduler_json_config_location =	(StrOpt) Absolute path to scheduler configuration JSON file.
scheduler_manager = *nova.scheduler.manager.SchedulerManager*	(StrOpt) Full class name for the Manager for scheduler
scheduler_max_attempts = *3*	(IntOpt) Maximum number of attempts to schedule an instance
scheduler_topic = *scheduler*	(StrOpt) The topic scheduler nodes listen on
scheduler_tracks_instance_changes = *True*	(BoolOpt) Determines if the Scheduler tracks changes to instances to help with its filtering decisions.
scheduler_use_baremetal_filters = *False*	(BoolOpt) Flag to decide whether to use baremetal_scheduler_default_filters or not.
scheduler_weight_classes = *nova.scheduler.weights.all_weighers*	(ListOpt) Which weight class names to use for weighing hosts
[cells]	
ram_weight_multiplier = *10.0*	(FloatOpt) Multiplier used for weighing ram. Negative numbers mean to stack vs spread.
scheduler_filter_classes = *nova.cells.filters.all_filters*	(ListOpt) Filter classes the cells scheduler should use. An entry of "nova.cells.filters.all_filters" maps to all cells filters included with nova.
scheduler_retries = *10*	(IntOpt) How many retries when no cells are available.
scheduler_retry_delay = *2*	(IntOpt) How often to retry in seconds when no cells are available.
scheduler_weight_classes = *nova.cells.weights.all_weighers*	(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]	
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 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.
weight_multiplier = *1.0*	(FloatOpt) Multiplier used for weighing 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.
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

Configuration option = Default value	Description
	ratio. So for "name1=1.0, name2=-1.0" The final weight would be name1.value * 1.0 + name2.value * -1.0.

Table 3.53. Description of serial console configuration options

Configuration option = Default value	Description
[serial_console]	
base_url = *ws://127.0.0.1:6083/*	(StrOpt) Location of serial console proxy.
enabled = *False*	(BoolOpt) Enable serial console related features
listen = *127.0.0.1*	(StrOpt) IP address on which instance serial console should listen
port_range = *10000:20000*	(StrOpt) Range of TCP ports to use for serial ports on compute hosts
proxyclient_address = *127.0.0.1*	(StrOpt) The address to which proxy clients (like nova-serialproxy) should connect
serialproxy_host = *0.0.0.0*	(StrOpt) Host on which to listen for incoming requests
serialproxy_port = *6083*	(IntOpt) Port on which to listen for incoming requests

Table 3.54. Description of SPICE configuration options

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"
html5proxy_host = *0.0.0.0*	(StrOpt) Host on which to listen for incoming requests
html5proxy_port = *6082*	(IntOpt) Port on which to listen for incoming requests
keymap = *en-us*	(StrOpt) Keymap for spice
server_listen = *127.0.0.1*	(StrOpt) IP address on which instance spice server should listen
server_proxyclient_address = *127.0.0.1*	(StrOpt) The address to which proxy clients (like nova-spicehtml5proxy) should connect

Table 3.55. Description of testing configuration options

Configuration option = Default value	Description
[DEFAULT]	
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
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 3.56. Description of trusted computing configuration options

Configuration option = Default value	Description
[trusted_computing]	

Configuration option = Default value	Description
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_insecure_ssl = *False*	(BoolOpt) Disable SSL cert verification for Attestation service
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 3.57. Description of upgrade levels configuration options

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 3.58. Description of VMware configuration options

Configuration option = Default value	Description
[vmware]	
api_retry_count = *10*	(IntOpt) The number of times we retry on failures, e.g., socket error, etc.
cache_prefix = *None*	(StrOpt) The prefix for Where cached images are stored. This is NOT the full path - just a folder prefix. This should only be used when a datastore cache should be shared between compute nodes. Note: this should only be used when the compute nodes have a shared file system.
cluster_name = *None*	(MultiStrOpt) Name of a VMware Cluster ComputeResource.
datastore_regex = *None*	(StrOpt) Regex to match the name of a datastore.

Configuration option = Default value	Description
host_ip = *None*	(StrOpt) Hostname or IP address for connection to VMware VC host.
host_password = *None*	(StrOpt) Password for connection to VMware VC host.
host_port = *443*	(IntOpt) Port for connection to VMware VC host.
host_username = *None*	(StrOpt) Username for connection to VMware 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.
pbm_default_policy = *None*	(StrOpt) The PBM default policy. If pbm_wsdl_location is set and there is no defined storage policy for the specific request then this policy will be used.
pbm_enabled = *False*	(BoolOpt) The PBM status.
pbm_wsdl_location = *None*	(StrOpt) PBM service WSDL file location URL. e.g. file:///opt/SDK/spbm/wsdl/pbmService.wsdl Not setting this will disable storage policy based placement of instances.
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 3.59. Description of VNC configuration options

Configuration option = Default value	Description
[DEFAULT]	
daemon = *False*	(BoolOpt) Become a daemon (background process)
key = *None*	(StrOpt) SSL key file (if separate from cert)
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"
novncproxy_host = *0.0.0.0*	(StrOpt) Host on which to listen for incoming requests
novncproxy_port = *6080*	(IntOpt) Port on which to listen for incoming requests
record = *False*	(BoolOpt) Record sessions to FILE.[session_number]
source_is_ipv6 = *False*	(BoolOpt) Source is ipv6
ssl_only = *False*	(BoolOpt) Disallow non-encrypted connections
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
web = */usr/share/spice-html5*	(StrOpt) Run webserver on same port. Serve files from DIR.
[vmware]	
vnc_port = *5900*	(IntOpt) VNC starting port
vnc_port_total = *10000*	(IntOpt) Total number of VNC ports

Table 3.60. Description of volumes configuration options

Configuration option = Default value	Description
[DEFAULT]	
block_device_allocate_retries = 60	(IntOpt) Number of times to retry block device allocation on failures
block_device_allocate_retries_interval = 3	(IntOpt) Waiting time interval (seconds) between block device allocation retries on failures
my_block_storage_ip = $my_ip	(StrOpt) Block storage IP address of this host
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
[cinder]	
cafile = None	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
catalog_info = volumev2:cinderv2: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>
certfile = None	(StrOpt) PEM encoded client certificate cert file
cross_az_attach = True	(BoolOpt) Allow attach between instance and volume in different availability zones.
endpoint_template = None	(StrOpt) Override service catalog lookup with template for cinder endpoint e.g. http://localhost:8776/v1/%(project_id)s
http_retries = 3	(IntOpt) Number of cinderclient retries on failed http calls
insecure = False	(BoolOpt) Verify HTTPS connections.
keyfile = None	(StrOpt) PEM encoded client certificate key file
os_region_name = None	(StrOpt) Region name of this node
timeout = None	(IntOpt) Timeout value for http requests
[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_iser_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

Configuration option = Default value	Description
scality_sofs_mount_point = *$state_path/scality*	(StrOpt) Base dir where Scality SOFS shall be mounted
smbfs_mount_options =	(StrOpt) Mount options passed to the SMBFS client. See mount.cifs man page for details. Note that the libvirt-qemu uid and gid must be specified.
smbfs_mount_point_base = *$state_path/mnt*	(StrOpt) Directory where the SMBFS shares are mounted on the compute node
[xenserver]	
block_device_creation_timeout = *10*	(IntOpt) Time to wait for a block device to be created

Table 3.61. Description of VPN configuration options

Configuration option = Default value	Description
[DEFAULT]	
boot_script_template = *$pybasedir/nova/cloudpipe/bootscript.template*	(StrOpt) Template for cloudpipe instance boot script
dmz_cidr =	(ListOpt) A list of dmz ranges that should be accepted
dmz_mask = *255.255.255.0*	(StrOpt) Netmask to push into openvpn config
dmz_net = *10.0.0.0*	(StrOpt) Network to push into openvpn config
vpn_flavor = *m1.tiny*	(StrOpt) Flavor for vpn instances
vpn_image_id = *0*	(StrOpt) Image ID used when starting up a cloudpipe vpn server
vpn_ip = *$my_ip*	(StrOpt) Public IP for the cloudpipe VPN servers
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 3.62. Description of Xen configuration options

Configuration option = Default value	Description
[DEFAULT]	
console_driver = *nova.console.xvp.XVPConsoleProxy*	(StrOpt) Driver to use for the console proxy
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_hvmloader_path = */usr/lib/xen/boot/hvmloader*	(StrOpt) Location where the Xen hvmloader is kept
[xenserver]	
agent_path = *usr/sbin/xe-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

Configuration option = Default value	Description
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 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. if <=0, should try once and no retry
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

Configuration option = Default value	Description
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-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
vif_driver = nova.virt.xenapi.vif.XenAPIBridgeDriver	(StrOpt) The XenAPI VIF driver using XenServer Network APIs.

Table 3.63. Description of XCP VNC proxy configuration options

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 3.64. Description of Zookeeper configuration options

Configuration option = Default value	Description
[zookeeper]	

Configuration option = Default value	Description
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
/: ec2cloud

[composite:ec2cloud]
use = call:nova.api.auth:pipeline_factory
noauth = ec2faultwrap logrequest ec2noauth cloudrequest validator ec2executor
noauth2 = 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]
paste.filter_factory = nova.api.ec2:Validator.factory

[app:ec2executor]
paste.app_factory = nova.api.ec2:Executor.factory

##############
# OpenStack #
##############

[composite:osapi_compute]
use = call:nova.api.openstack.urlmap:urlmap_factory
/: oscomputeversions
/v1.1: openstack_compute_api_v2
/v2: openstack_compute_api_v2
/v2.1: openstack_compute_api_v21
/v3: openstack_compute_api_v3

[composite:openstack_compute_api_v2]
use = call:nova.api.auth:pipeline_factory
noauth = compute_req_id faultwrap sizelimit noauth ratelimit
 osapi_compute_app_v2
noauth2 = compute_req_id faultwrap sizelimit noauth2 ratelimit
 osapi_compute_app_v2
keystone = compute_req_id faultwrap sizelimit authtoken keystonecontext
 ratelimit osapi_compute_app_v2
keystone_nolimit = compute_req_id faultwrap sizelimit authtoken
 keystonecontext osapi_compute_app_v2

[composite:openstack_compute_api_v21]
use = call:nova.api.auth:pipeline_factory_v21
noauth = compute_req_id faultwrap sizelimit noauth osapi_compute_app_v21
noauth2 = compute_req_id faultwrap sizelimit noauth2 osapi_compute_app_v21
keystone = compute_req_id faultwrap sizelimit authtoken keystonecontext
 osapi_compute_app_v21

[composite:openstack_compute_api_v3]
use = call:nova.api.auth:pipeline_factory_v21
noauth = request_id faultwrap sizelimit noauth_v3 osapi_compute_app_v3
noauth2 = request_id faultwrap sizelimit noauth_v3 osapi_compute_app_v3
keystone = request_id faultwrap sizelimit authtoken keystonecontext
 osapi_compute_app_v3

[filter:request_id]
paste.filter_factory = oslo.middleware:RequestId.factory

[filter:compute_req_id]
paste.filter_factory = nova.api.compute_req_id:ComputeReqIdMiddleware.factory

[filter:faultwrap]
```

```
paste.filter_factory = nova.api.openstack:FaultWrapper.factory

[filter:noauth]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddlewareOld.factory

[filter:noauth2]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddleware.factory

[filter:noauth_v3]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddlewareV3.factory

[filter:ratelimit]
paste.filter_factory = nova.api.openstack.compute.
limits:RateLimitingMiddleware.factory

[filter:sizelimit]
paste.filter_factory = oslo.middleware:RequestBodySizeLimiter.factory

[app:osapi_compute_app_v2]
paste.app_factory = nova.api.openstack.compute:APIRouter.factory

[app:osapi_compute_app_v21]
paste.app_factory = nova.api.openstack.compute:APIRouterV21.factory

[app:osapi_compute_app_v3]
paste.app_factory = nova.api.openstack.compute:APIRouterV3.factory

[pipeline:oscomputeversions]
pipeline = faultwrap oscomputeversionapp

[app:oscomputeversionapp]
paste.app_factory = nova.api.openstack.compute.versions:Versions.factory

##########
# Shared #
##########

[filter:keystonecontext]
paste.filter_factory = nova.api.auth:NovaKeystoneContext.factory

[filter:authtoken]
paste.filter_factory = keystonemiddleware.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:resize": "",
    "compute:confirm_resize": "",
    "compute:revert_resize": "",
    "compute:rebuild": "",
    "compute:reboot": "",

    "compute:volume_snapshot_create": "",
    "compute:volume_snapshot_delete": "",

    "admin_api": "is_admin:True",
    "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:aggregates": "rule:admin_api",
    "compute_extension:agents": "rule:admin_api",
    "compute_extension:attach_interfaces": "",
    "compute_extension:baremetal_nodes": "rule:admin_api",
    "compute_extension:cells": "rule:admin_api",
    "compute_extension:cells:create": "rule:admin_api",
    "compute_extension:cells:delete": "rule:admin_api",
    "compute_extension:cells:update": "rule:admin_api",
    "compute_extension:cells:sync_instances": "rule:admin_api",
    "compute_extension:certificates": "",
    "compute_extension:cloudpipe": "rule:admin_api",
    "compute_extension:cloudpipe_update": "rule:admin_api",
    "compute_extension:console_output": "",
    "compute_extension:consoles": "",
    "compute_extension:createserverext": "",
    "compute_extension:deferred_delete": "",
    "compute_extension:disk_config": "",
    "compute_extension:evacuate": "rule:admin_api",
    "compute_extension:extended_server_attributes": "rule:admin_api",
    "compute_extension:extended_status": "",
    "compute_extension:extended_availability_zone": "",
    "compute_extension:extended_ips": "",
    "compute_extension:extended_ips_mac": "",
    "compute_extension:extended_vif_net": "",
    "compute_extension:extended_volumes": "",
```

```
    "compute_extension:fixed_ips": "rule:admin_api",
    "compute_extension:flavor_access": "",
    "compute_extension:flavor_access:addTenantAccess": "rule:admin_api",
    "compute_extension:flavor_access:removeTenantAccess": "rule:admin_api",
    "compute_extension:flavor_disabled": "",
    "compute_extension:flavor_rxtx": "",
    "compute_extension:flavor_swap": "",
    "compute_extension:flavorextradata": "",
    "compute_extension:flavorextraspecs:index": "",
    "compute_extension:flavorextraspecs:show": "",
    "compute_extension:flavorextraspecs:create": "rule:admin_api",
    "compute_extension:flavorextraspecs:update": "rule:admin_api",
    "compute_extension:flavorextraspecs:delete": "rule:admin_api",
    "compute_extension:flavormanage": "rule:admin_api",
    "compute_extension:floating_ip_dns": "",
    "compute_extension:floating_ip_pools": "",
    "compute_extension:floating_ips": "",
    "compute_extension:floating_ips_bulk": "rule:admin_api",
    "compute_extension:fping": "",
    "compute_extension:fping:all_tenants": "rule:admin_api",
    "compute_extension:hide_server_addresses": "is_admin:False",
    "compute_extension:hosts": "rule:admin_api",
    "compute_extension:hypervisors": "rule:admin_api",
    "compute_extension:image_size": "",
    "compute_extension:instance_actions": "",
    "compute_extension:instance_actions:events": "rule:admin_api",
    "compute_extension:instance_usage_audit_log": "rule:admin_api",
    "compute_extension:keypairs": "",
    "compute_extension:keypairs:index": "",
    "compute_extension:keypairs:show": "",
    "compute_extension:keypairs:create": "",
    "compute_extension:keypairs:delete": "",
    "compute_extension: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:quota_classes": "",
    "compute_extension:rescue": "",
    "compute_extension:security_group_default_rules": "rule:admin_api",
    "compute_extension:security_groups": "",
    "compute_extension:server_diagnostics": "rule:admin_api",
    "compute_extension:server_groups": "",
    "compute_extension:server_password": "",
    "compute_extension:server_usage": "",
    "compute_extension:services": "rule:admin_api",
    "compute_extension:shelve": "",
    "compute_extension:shelveOffload": "rule:admin_api",
    "compute_extension:simple_tenant_usage:show": "rule:admin_or_owner",
    "compute_extension:simple_tenant_usage:list": "rule:admin_api",
    "compute_extension: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:availability_zone:detail": "rule:admin_api",
    "compute_extension:used_limits_for_admin": "rule:admin_api",
    "compute_extension:migrations:index": "rule:admin_api",
    "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:os-server-external-events:create": "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: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": "",
    "network:attach_external_network": "rule:admin_api",

    "os_compute_api:servers:start": "rule:admin_or_owner",
    "os_compute_api:servers:stop": "rule:admin_or_owner",
    "os_compute_api:os-access-ips:discoverable": "",
```

```
    "os_compute_api:os-access-ips": "",
    "os_compute_api:os-admin-actions": "rule:admin_api",
    "os_compute_api:os-admin-actions:discoverable": "",
    "os_compute_api:os-admin-actions:reset_network": "rule:admin_api",
    "os_compute_api:os-admin-actions:inject_network_info": "rule:admin_api",
    "os_compute_api:os-admin-actions:reset_state": "rule:admin_api",
    "os_compute_api:os-admin-password": "",
    "os_compute_api:os-admin-password:discoverable": "",
    "os_compute_api:os-aggregates:discoverable": "",
    "os_compute_api:os-aggregates:index": "rule:admin_api",
    "os_compute_api:os-aggregates:create": "rule:admin_api",
    "os_compute_api:os-aggregates:show": "rule:admin_api",
    "os_compute_api:os-aggregates:update": "rule:admin_api",
    "os_compute_api:os-aggregates:delete": "rule:admin_api",
    "os_compute_api:os-aggregates:add_host": "rule:admin_api",
    "os_compute_api:os-aggregates:remove_host": "rule:admin_api",
    "os_compute_api:os-aggregates:set_metadata": "rule:admin_api",
    "os_compute_api:os-agents": "rule:admin_api",
    "os_compute_api:os-agents:discoverable": "",
    "os_compute_api:os-attach-interfaces": "",
    "os_compute_api:os-attach-interfaces:discoverable": "",
    "os_compute_api:os-baremetal-nodes": "rule:admin_api",
    "os_compute_api:os-baremetal-nodes:discoverable": "",
    "os_compute_api:os-block-device-mapping-v1:discoverable": "",
    "os_compute_api:os-cells": "rule:admin_api",
    "os_compute_api:os-cells:create": "rule:admin_api",
    "os_compute_api:os-cells:delete": "rule:admin_api",
    "os_compute_api:os-cells:update": "rule:admin_api",
    "os_compute_api:os-cells:sync_instances": "rule:admin_api",
    "os_compute_api:os-cells:discoverable": "",
    "os_compute_api:os-certificates:create": "",
    "os_compute_api:os-certificates:show": "",
    "os_compute_api:os-certificates:discoverable": "",
    "os_compute_api:os-cloudpipe": "rule:admin_api",
    "os_compute_api:os-cloudpipe:discoverable": "",
    "os_compute_api:os-consoles:discoverable": "",
    "os_compute_api:os-consoles:create": "",
    "os_compute_api:os-consoles:delete": "",
    "os_compute_api:os-consoles:index": "",
    "os_compute_api:os-consoles:show": "",
    "os_compute_api:os-console-output:discoverable": "",
    "os_compute_api:os-console-output": "",
    "os_compute_api:os-remote-consoles": "",
    "os_compute_api:os-remote-consoles:discoverable": "",
    "os_compute_api:os-create-backup:discoverable": "",
    "os_compute_api:os-create-backup": "rule:admin_or_owner",
    "os_compute_api:os-deferred-delete": "",
    "os_compute_api:os-deferred-delete:discoverable": "",
    "os_compute_api:os-disk-config": "",
    "os_compute_api:os-disk-config:discoverable": "",
    "os_compute_api:os-evacuate": "rule:admin_api",
    "os_compute_api:os-evacuate:discoverable": "",
    "os_compute_api:os-extended-server-attributes": "rule:admin_api",
    "os_compute_api:os-extended-server-attributes:discoverable": "",
    "os_compute_api:os-extended-status": "",
    "os_compute_api:os-extended-status:discoverable": "",
    "os_compute_api:os-extended-availability-zone": "",
    "os_compute_api:os-extended-availability-zone:discoverable": "",
    "os_compute_api:extension_info:discoverable": "",
    "os_compute_api:os-extended-volumes": "",
```

```
    "os_compute_api:os-extended-volumes:discoverable": "",
    "os_compute_api:os-fixed-ips": "rule:admin_api",
    "os_compute_api:os-fixed-ips:discoverable": "",
    "os_compute_api:os-flavor-access": "",
    "os_compute_api:os-flavor-access:discoverable": "",
    "os_compute_api:os-flavor-access:remove_tenant_access": "rule:admin_api",
    "os_compute_api:os-flavor-access:add_tenant_access": "rule:admin_api",
    "os_compute_api:os-flavor-rxtx": "",
    "os_compute_api:os-flavor-rxtx:discoverable": "",
    "os_compute_api:flavors:discoverable": "",
    "os_compute_api:os-flavor-extra-specs:discoverable": "",
    "os_compute_api:os-flavor-extra-specs:index": "",
    "os_compute_api:os-flavor-extra-specs:show": "",
    "os_compute_api:os-flavor-extra-specs:create": "rule:admin_api",
    "os_compute_api:os-flavor-extra-specs:update": "rule:admin_api",
    "os_compute_api:os-flavor-extra-specs:delete": "rule:admin_api",
    "os_compute_api:os-flavor-manage:discoverable": "",
    "os_compute_api:os-flavor-manage": "rule:admin_api",
    "os_compute_api:os-floating-ip-dns": "",
    "os_compute_api:os-floating-ip-dns:discoverable": "",
    "os_compute_api:os-floating-ip-pools": "",
    "os_compute_api:os-floating-ip-pools:discoverable": "",
    "os_compute_api:os-floating-ips": "",
    "os_compute_api:os-floating-ips:discoverable": "",
    "os_compute_api:os-floating-ips-bulk": "rule:admin_api",
    "os_compute_api:os-floating-ips-bulk:discoverable": "",
    "os_compute_api:os-fping": "",
    "os_compute_api:os-fping:discoverable": "",
    "os_compute_api:os-fping:all_tenants": "rule:admin_api",
    "os_compute_api:os-hide-server-addresses": "is_admin:False",
    "os_compute_api:os-hide-server-addresses:discoverable": "",
    "os_compute_api:os-hosts": "rule:admin_api",
    "os_compute_api:os-hosts:discoverable": "",
    "os_compute_api:os-hypervisors": "rule:admin_api",
    "os_compute_api:os-hypervisors:discoverable": "",
    "os_compute_api:images:discoverable": "",
    "os_compute_api:image-size": "",
    "os_compute_api:image-size:discoverable": "",
    "os_compute_api:os-instance-actions": "",
    "os_compute_api:os-instance-actions:discoverable": "",
    "os_compute_api:os-instance-actions:events": "rule:admin_api",
    "os_compute_api:os-instance-usage-audit-log": "rule:admin_api",
    "os_compute_api:os-instance-usage-audit-log:discoverable": "",
    "os_compute_api:ips:discoverable": "",
    "os_compute_api:ips:index": "rule:admin_or_owner",
    "os_compute_api:ips:show": "rule:admin_or_owner",
    "os_compute_api:os-keypairs:discoverable": "",
    "os_compute_api:os-keypairs": "",
    "os_compute_api:os-keypairs:index": "",
    "os_compute_api:os-keypairs:show": "",
    "os_compute_api:os-keypairs:create": "",
    "os_compute_api:os-keypairs:delete": "",
    "os_compute_api:limits:discoverable": "",
    "os_compute_api:os-lock-server:discoverable": "",
    "os_compute_api:os-lock-server:lock": "rule:admin_or_owner",
    "os_compute_api:os-lock-server:unlock": "rule:admin_or_owner",
    "os_compute_api:os-migrate-server:discoverable": "",
    "os_compute_api:os-migrate-server:migrate": "rule:admin_api",
    "os_compute_api:os-migrate-server:migrate_live": "rule:admin_api",
    "os_compute_api:os-multinic": "",
```

```
    "os_compute_api:os-multinic:discoverable": "",
    "os_compute_api:os-networks": "rule:admin_api",
    "os_compute_api:os-networks:view": "",
    "os_compute_api:os-networks:discoverable": "",
    "os_compute_api:os-networks-associate": "rule:admin_api",
    "os_compute_api:os-networks-associate:discoverable": "",
    "os_compute_api:os-pause-server:discoverable": "",
    "os_compute_api:os-pause-server:pause": "rule:admin_or_owner",
    "os_compute_api:os-pause-server:unpause": "rule:admin_or_owner",
    "os_compute_api:os-pci:pci_servers": "",
    "os_compute_api:os-pci:discoverable": "",
    "os_compute_api:os-pci:index": "rule:admin_api",
    "os_compute_api:os-pci:detail": "rule:admin_api",
    "os_compute_api:os-pci:show": "rule:admin_api",
    "os_compute_api:os-personality:discoverable": "",
    "os_compute_api:os-preserve-ephemeral-rebuild:discoverable": "",
    "os_compute_api:os-quota-sets:discoverable": "",
    "os_compute_api:os-quota-sets:show": "",
    "os_compute_api:os-quota-sets:update": "rule:admin_api",
    "os_compute_api:os-quota-sets:delete": "rule:admin_api",
    "os_compute_api:os-quota-sets:detail": "rule:admin_api",
    "os_compute_api:os-quota-class-sets": "",
    "os_compute_api:os-quota-class-sets:discoverable": "",
    "os_compute_api:os-rescue": "",
    "os_compute_api:os-rescue:discoverable": "",
    "os_compute_api:os-scheduler-hints:discoverable": "",
    "os_compute_api:os-security-group-default-rules:discoverable": "",
    "os_compute_api:os-security-group-default-rules": "rule:admin_api",
    "os_compute_api:os-security-groups": "",
    "os_compute_api:os-security-groups:discoverable": "",
    "os_compute_api:os-server-diagnostics": "rule:admin_api",
    "os_compute_api:os-server-diagnostics:discoverable": "",
    "os_compute_api:os-server-password": "",
    "os_compute_api:os-server-password:discoverable": "",
    "os_compute_api:os-server-usage": "",
    "os_compute_api:os-server-usage:discoverable": "",
    "os_compute_api:os-server-groups": "",
    "os_compute_api:os-server-groups:discoverable": "",
    "os_compute_api:os-services": "rule:admin_api",
    "os_compute_api:os-services:discoverable": "",
    "os_compute_api:server-metadata:discoverable": "",
    "os_compute_api:server-metadata:index": "rule:admin_or_owner",
    "os_compute_api:server-metadata:show": "rule:admin_or_owner",
    "os_compute_api:server-metadata:delete": "rule:admin_or_owner",
    "os_compute_api:server-metadata:create": "rule:admin_or_owner",
    "os_compute_api:server-metadata:update": "rule:admin_or_owner",
    "os_compute_api:server-metadata:update_all": "rule:admin_or_owner",
    "os_compute_api:servers:discoverable": "",
    "os_compute_api:os-shelve:shelve": "",
    "os_compute_api:os-shelve:shelve:discoverable": "",
    "os_compute_api:os-shelve:shelve_offload": "rule:admin_api",
    "os_compute_api:os-simple-tenant-usage:discoverable": "",
    "os_compute_api:os-simple-tenant-usage:show": "rule:admin_or_owner",
    "os_compute_api:os-simple-tenant-usage:list": "rule:admin_api",
    "os_compute_api:os-suspend-server:discoverable": "",
    "os_compute_api:os-suspend-server:suspend": "rule:admin_or_owner",
    "os_compute_api:os-suspend-server:resume": "rule:admin_or_owner",
    "os_compute_api:os-tenant-networks": "rule:admin_or_owner",
    "os_compute_api:os-tenant-networks:discoverable": "",
    "os_compute_api:os-shelve:unshelve": "",
```

```
    "os_compute_api:os-user-data:discoverable": "",
    "os_compute_api:os-virtual-interfaces": "",
    "os_compute_api:os-virtual-interfaces:discoverable": "",
    "os_compute_api:os-volumes": "",
    "os_compute_api:os-volumes:discoverable": "",
    "os_compute_api:os-volumes-attachments:index": "",
    "os_compute_api:os-volumes-attachments:show": "",
    "os_compute_api:os-volumes-attachments:create": "",
    "os_compute_api:os-volumes-attachments:update": "",
    "os_compute_api:os-volumes-attachments:delete": "",
    "os_compute_api:os-volumes-attachments:discoverable": "",
    "os_compute_api:os-availability-zone:list": "",
    "os_compute_api:os-availability-zone:discoverable": "",
    "os_compute_api:os-availability-zone:detail": "rule:admin_api",
    "os_compute_api:os-used-limits": "rule:admin_api",
    "os_compute_api:os-used-limits:discoverable": "",
    "os_compute_api:os-migrations:index": "rule:admin_api",
    "os_compute_api:os-migrations:discoverable": "",
    "os_compute_api:os-assisted-volume-snapshots:create": "rule:admin_api",
    "os_compute_api:os-assisted-volume-snapshots:delete": "rule:admin_api",
    "os_compute_api:os-assisted-volume-snapshots:discoverable": "",
    "os_compute_api:os-console-auth-tokens": "rule:admin_api",
    "os_compute_api:os-server-external-events:create": "rule:admin_api"
}
```

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.

It is also possible to disable the root wrapper, and default to sudo only. Configure the *`disable_rootwrap`* option in the `[workaround]` section of the `nova.conf` configuration file.

```
                # 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
# explicitely 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
```

New, updated and deprecated options in Kilo for OpenStack Compute

Table 3.65. New options

Option = default value	(Type) Help string
[DEFAULT] ebtables_exec_attempts = 3	(IntOpt) Number of times to retry ebtables commands on failure.
[DEFAULT] ebtables_retry_interval = 1.0	(FloatOpt) Number of seconds to wait between ebtables retries.
[DEFAULT] io_ops_weight_multiplier = -1.0	(FloatOpt) Multiplier used for weighing host io ops. Negative numbers mean a preference to choose light workload compute hosts.
[DEFAULT] keystone_ec2_insecure = False	(BoolOpt) Disable SSL certificate verification.
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] max_concurrent_builds = 10	(IntOpt) Maximum number of instance builds to run concurrently
[DEFAULT] metadata_cache_expiration = 15	(IntOpt) Time in seconds to cache metadata; 0 to disable metadata caching entirely (not recommended). Increasingthis should improve response times of the metadata API when under heavy load. Higher values may increase memoryusage and result in longer times for host metadata changes to take effect.
[DEFAULT] my_block_storage_ip = $my_ip	(StrOpt) Block storage IP address of this host
[DEFAULT] novncproxy_host = 0.0.0.0	(StrOpt) Host on which to listen for incoming requests
[DEFAULT] novncproxy_port = 6080	(IntOpt) Port on which to listen for incoming requests
[DEFAULT] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[DEFAULT] quota_networks = 3	(IntOpt) Number of private networks allowed per project

Option = default value	(Type) Help string
[DEFAULT] scheduler_instance_sync_interval = 120	(IntOpt) Waiting time interval (seconds) between sending the scheduler a list of current instance UUIDs to verify that its view of instances is in sync with nova. If the CONF option `scheduler_tracks_instance_changes` is False, changing this option will have no effect.
[DEFAULT] scheduler_tracks_instance_changes = True	(BoolOpt) Determines if the Scheduler tracks changes to instances to help with its filtering decisions.
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
[api_database] connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the Nova API database.
[api_database] connection_debug = 0	(IntOpt) Verbosity of SQL debugging information: 0=None, 100=Everything.
[api_database] connection_trace = False	(BoolOpt) Add Python stack traces to SQL as comment strings.
[api_database] idle_timeout = 3600	(IntOpt) Timeout before idle SQL connections are reaped.
[api_database] max_overflow = None	(IntOpt) If set, use this value for max_overflow with SQLAlchemy.
[api_database] max_pool_size = None	(IntOpt) Maximum number of SQL connections to keep open in a pool.
[api_database] max_retries = 10	(IntOpt) Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
[api_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=
[api_database] pool_timeout = None	(IntOpt) If set, use this value for pool_timeout with SQLAlchemy.
[api_database] retry_interval = 10	(IntOpt) Interval between retries of opening a SQL connection.
[api_database] slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
[api_database] sqlite_synchronous = True	(BoolOpt) If True, SQLite uses synchronous mode.
[barbican] cafile = None	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
[barbican] catalog_info = key-manager:barbican:public	(StrOpt) Info to match when looking for barbican in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type>
[barbican] certfile = None	(StrOpt) PEM encoded client certificate cert file
[barbican] endpoint_template = None	(StrOpt) Override service catalog lookup with template for barbican endpoint e.g. http://localhost:9311/v1/%(project_id)s
[barbican] insecure = False	(BoolOpt) Verify HTTPS connections.
[barbican] keyfile = None	(StrOpt) PEM encoded client certificate key file
[barbican] os_region_name = None	(StrOpt) Region name of this node
[barbican] timeout = None	(IntOpt) Timeout value for http requests

Option = default value	(Type) Help string
[cinder] cafile = None	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
[cinder] certfile = None	(StrOpt) PEM encoded client certificate cert file
[cinder] insecure = False	(BoolOpt) Verify HTTPS connections.
[cinder] keyfile = None	(StrOpt) PEM encoded client certificate key file
[cinder] timeout = None	(IntOpt) Timeout value for http requests
[database] backend = sqlalchemy	(StrOpt) The back end to use for the database.
[database] connection = None	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
[database] db_max_retries = 20	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
[database] db_max_retry_interval = 10	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
[database] db_retry_interval = 1	(IntOpt) Seconds between retries of a database transaction.
[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 number of database connection retries during startup. Set to -1 to specify 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] slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
[database] sqlite_db = oslo.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.
[guestfs] debug = False	(BoolOpt) Enable guestfs debug
[libvirt] iscsi_iface = None	(StrOpt) The iSCSI transport iface to use to connect to target in case offload support is desired. Supported transports are be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx and ocs. Default format is transport_name.hwaddress and can be generated manually or via iscsiadm -m iface

Option = default value	(Type) Help string
[libvirt] quobyte_client_cfg = None	(StrOpt) Path to a Quobyte Client configuration file.
[libvirt] quobyte_mount_point_base = $state_path/mnt	(StrOpt) Directory where the Quobyte volume is mounted on the compute node
[libvirt] smbfs_mount_options =	(StrOpt) Mount options passed to the SMBFS client. See mount.cifs man page for details. Note that the libvirt-qemu uid and gid must be specified.
[libvirt] smbfs_mount_point_base = $state_path/mnt	(StrOpt) Directory where the SMBFS shares are mounted on the compute node
[neutron] auth_plugin = None	(StrOpt) Name of the plugin to load
[neutron] auth_section = None	(StrOpt) Config Section from which to load plugin specific options
[neutron] cafile = None	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
[neutron] certfile = None	(StrOpt) PEM encoded client certificate cert file
[neutron] insecure = False	(BoolOpt) Verify HTTPS connections.
[neutron] keyfile = None	(StrOpt) PEM encoded client certificate key file
[neutron] timeout = None	(IntOpt) Timeout value for http requests
[oslo_concurrency] disable_process_locking = False	(BoolOpt) Enables or disables inter-process locks.
[oslo_concurrency] lock_path = None	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.

Option = default value	(Type) Help string
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.
[serial_console] serialproxy_host = 0.0.0.0	(StrOpt) Host on which to listen for incoming requests
[serial_console] serialproxy_port = 6083	(IntOpt) Port on which to listen for incoming requests
[vmware] cache_prefix = None	(StrOpt) The prefix for Where cached images are stored. This is NOT the full path - just a folder prefix. This should only be used when a datastore cache should be shared between compute nodes. Note: this should only be used when the compute nodes have a shared file system.
[vmware] pbm_default_policy = None	(StrOpt) The PBM default policy. If pbm_wsdl_location is set and there is no defined storage policy for the specific request then this policy will be used.
[vmware] pbm_enabled = False	(BoolOpt) The PBM status.
[vmware] pbm_wsdl_location = None	(StrOpt) PBM service WSDL file location URL. e.g. file:///opt/SDK/spbm/wsdl/pbmService.wsdl Not setting this will disable storage policy based placement of instances.
[workarounds] destroy_after_evacuate = True	(BoolOpt) Whether to destroy instances on startup when we suspect they have previously been evacuated. This can result in data loss if undesired. See https://launchpad.net/bugs/1419785
[workarounds] disable_libvirt_livesnapshot = True	(BoolOpt) When using libvirt 1.2.2 fails live snapshots intermittently under load. This config option provides mechanism to disable livesnapshot while this is resolved. See https://bugs.launchpad.net/nova/+bug/1334398
[workarounds] disable_rootwrap = False	(BoolOpt) This option allows a fallback to sudo for performance reasons. For example see https://bugs.launchpad.net/nova/+bug/1415106

Table 3.66. New default values

Option	Previous default value	New default value
[DEFAULT] client_socket_timeout	0	900
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] ec2_path	/services/Cloud	/
[DEFAULT] multi_instance_display_name_template	%(name)s-%(uuid)s	%(name)s-%(count)d
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	local
[cinder] catalog_info	volume:cinder:publicURL	volumev2:cinderv2:publicURL

Table 3.67. Deprecated options

Deprecated option	New Option
[DEFAULT] network_device_mtu	None
[DEFAULT] log-format	None
[DEFAULT] use-syslog	None
[cinder] http_timeout	[cinder] timeout
[DEFAULT] use_syslog	None
[ironic] client_log_level	None
[neutron] admin_username	None
[DEFAULT] osapi_max_request_body_size	[oslo_middleware] max_request_body_size
[neutron] ca_certificates_file	[neutron] cafile
[neutron] auth_strategy	None
[neutron] admin_user_id	None
[neutron] admin_tenant_id	None
[DEFAULT] log_format	None
[cinder] api_insecure	[cinder] insecure
[neutron] admin_tenant_name	None
[neutron] admin_password	None
[DEFAULT] share_dhcp_address	None
[neutron] api_insecure	[neutron] insecure
[cinder] ca_certificates_file	[cinder] cafile
[neutron] admin_auth_url	None
[neutron] url_timeout	[neutron] timeout
[neutron] allow_duplicate_networks	None

4. Dashboard

Table of Contents

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 = 'elj1IWiLoWHgryYxFT6j7cM5fGOOxWY0'

# 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.

2. 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.

This example uses the `http://openstack.example.com` domain. Use a domain that fits your current setup.

1. In the `/etc/openstack-dashboard/local_settings.py` file, update the following options:

```
USE_SSL = True
CSRF_COOKIE_SECURE = True
SESSION_COOKIE_SECURE = True
SESSION_COOKIE_HTTPONLY = True
```

To enable HTTPS, the `USE_SSL = True` option is required.

The other options require that HTTPS is enabled; these options defend against cross-site scripting.

2. Edit the `/etc/apache2/conf.d/openstack-dashboard.conf` file as shown in Example 4.2, “After” [338]:

Example 4.1. 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>
```

Example 4.2. 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, the Apache HTTP server listens on port 443 and redirects all non-secure requests to the HTTPS protocol. The secured section defines the private key, public key, and certificate to use.

3. Restart the Apache HTTP server.

   For Debian, Ubuntu, or SUSE distributions:

   ```
   # service apache2 restart
   ```

   For Fedora, RHEL, or CentOS distributions:

   ```
   # service httpd restart
   ```

4. Restart `memcached`:

```
# service memcached restart
```

If you try to access the dashboard through HTTP, the browser redirects you to the HTTPS page.

Customize the dashboard

Once you have the dashboard installed you can customize the way it looks and feels to suit your own needs.

Note

The OpenStack dashboard by default on Ubuntu installs the `openstack-dashboard-ubuntu-theme` package.

If you do not want to use this theme you can remove it and its dependencies using the following command:

```
# apt-get remove --auto-remove openstack-dashboard-ubuntu-theme
```

Note

This guide focuses on the `local_settings.py` file, stored in `/openstack-dashboard/openstack_dashboard/local/`.

This guide is adapted from How To Custom Brand The OpenStack "Horizon" Dashboard.

The following can easily be customized:

- Site colors
- Logo
- HTML title
- Site branding link
- Help URL

Procedure 4.1. Logo and site colors

1. Create two logo files, png format, with transparent backgrounds using the following sizes:
   - Login screen: 365 x 50
   - Logged in banner: 216 x 35
2. Upload your new images to the following location: `/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/img/`
3. Create a CSS style sheet in the following directory: `/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/css/`

4. 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>
*/
h1.brand {
background: #355796 repeat-x top left;
border-bottom: 2px solid #BAD3E1;
}
h1.brand a {
background: url(../img/my_cloud_logo_small.png) top left no-repeat;
}
#splash .login {
background: #355796 url(../img/my_cloud_logo_medium.png) no-repeat center 35px;
}
#splash .login .modal-header {
border-top: 1px solid #BAD3E1;
}
.btn-primary {
background-image: none !important;
background-color: #355796 !important;
border: none !important;
box-shadow: none;
}
.btn-primary:hover,
.btn-primary:active {
border: none;
box-shadow: none;
background-color: #BAD3E1 !important;
text-decoration: none;
}
```

5. Open the following HTML template in an editor of your choice: `/usr/share/openstack-dashboard/openstack_dashboard/templates/_stylesheets.html`

6. Add a line to include your newly created style sheet. For example `custom.css` file:

```
...
    <link href='{{ STATIC_URL }}bootstrap/css/bootstrap.min.css' media='screen' rel=
'stylesheet' />
    <link href='{{ STATIC_URL }}dashboard/css/{% choose_css %}' media='screen' rel='stylesheet' /
>
    <link href='{{ STATIC_URL }}dashboard/css/custom.css' media='screen' rel='stylesheet' />
    ...
```

7. **Restart Apache:**

   On Ubuntu:

```
# service apache2 restart
```

   On Fedora, RHEL, CentOS:

```
# service httpd restart
```

   On openSUSE:

```
# service apache2 restart
```

8. To view your changes simply reload your dashboard. If necessary go back and modify your CSS file as appropriate.

Procedure 4.2. HTML title

1. Set the HTML title, which appears at the top of the browser window, by adding the following line to `local_settings.py`:

```
SITE_BRANDING = "Example, Inc. Cloud"
```

2. Restart Apache for this change to take effect.

Procedure 4.3. HTML title

1. The logo also acts as a hyperlink. The default behavior is to redirect to `horizon:user_home`. To change this, add the following attribute to `local_settings.py`

```
SITE_BRANDING_LINK = "http://example.com"
```

2. Restart Apache for this change to take effect.

Procedure 4.4. Help URL

1. By default the help URL points to http://docs.openstack.org. Change this by editing the following arritbute to the URL of your choice in `local_settings.py`

```
'help_url': "http://openstack.mycompany.org",
```

2. Restart Apache for this change to take effect.

Additional sample configuration files

Find the following files in `/etc/openstack-dashboard`.

keystone_policy.json

The `keystone_policy.json` file defines additional access controls for the dashboard that apply to the Identity service.

Note

The `keystone_policy.json` file must match the Identity service `/etc/keystone/policy.json` policy file.

```
{
    "admin_required": [
        [
            "role:admin"
        ],
        [
            "is_admin:1"
        ]
    ],
    "service_role": [
        [
            "role:service"
        ]
```

```
    ],
    "service_or_admin": [
        [
            "rule:admin_required"
        ],
        [
            "rule:service_role"
        ]
    ],
    "owner": [
        [
            "user_id:%(user_id)s"
        ]
    ],
    "admin_or_owner": [
        [
            "rule:admin_required"
        ],
        [
            "rule:owner"
        ]
    ],
    "default": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_service": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_services": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:create_service": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_service": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_service": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_endpoint": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_endpoints": [
        [
            "rule:admin_required"
```

```
        ]
    ],
    "identity:create_endpoint": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_endpoint": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_endpoint": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_domain": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_domains": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:create_domain": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_domain": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_domain": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_project": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_projects": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_user_projects": [
        [
            "rule:admin_or_owner"
        ]
    ],
    "identity:create_project": [
        [
            "rule:admin_required"
```

```
        ]
    ],
    "identity:update_project": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_project": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_user": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_users": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:create_user": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_user": [
        [
            "rule:admin_or_owner"
        ]
    ],
    "identity:delete_user": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_groups": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_groups_for_user": [
        [
            "rule:admin_or_owner"
        ]
    ],
    "identity:create_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_group": [
        [
            "rule:admin_required"
```

```
        ]
    ],
    "identity:delete_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_users_in_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:remove_user_from_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:check_user_in_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:add_user_to_group": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_credential": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_credentials": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:create_credential": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_credential": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_credential": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:get_role": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_roles": [
        [
            "rule:admin_required"
```

```
        ]
    ],
    "identity:create_role": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:update_role": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:delete_role": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:check_grant": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_grants": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:create_grant": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:revoke_grant": [
        [
            "rule:admin_required"
        ]
    ],
    "identity:list_role_assignments": [
        [
            "rule:admin_required"
        ]
    ],
    "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"
        ]
    ],
    "identity:list_trusts": [
        [
            "@"
        ]
    ],
    "identity:list_roles_for_trust": [
        [
            "@"
        ]
    ],
    "identity:check_role_for_trust": [
        [
            "@"
        ]
    ],
    "identity:get_role_for_trust": [
        [
            "@"
```

```
        ]
    ],
    "identity:delete_trust": [
        [
            "@"
        ]
    ]
}
```

nova_policy.json

The `nova_policy.json` file defines additional access controls for the dashboard that apply to the Compute service.

Note

The `nova_policy.json` file must match the Compute `/etc/nova/policy.json` policy file.

```
{
    "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:unlock_override": "rule:admin_api",
    "compute:shelve": "",
    "compute:shelve_offload": "",
    "compute:unshelve": "",
    "admin_api": "is_admin:True",
    "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:pause": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:unpause": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:suspend": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:resume": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:lock": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:unlock": "rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:reset_network": "rule:admin_api",
    "compute_extension:v3:os-admin-actions:inject_network_info":
"rule:admin_api",
```

```
    "compute_extension:v3:os-admin-actions:create_backup":
"rule:admin_or_owner",
    "compute_extension:v3:os-admin-actions:migrate_live": "rule:admin_api",
    "compute_extension:v3:os-admin-actions:reset_state": "rule:admin_api",
    "compute_extension:v3:os-admin-actions:migrate": "rule:admin_api",
    "compute_extension:v3:os-admin-password": "",
    "compute_extension:aggregates": "rule:admin_api",
    "compute_extension:v3:os-aggregates": "rule:admin_api",
    "compute_extension:agents": "rule:admin_api",
    "compute_extension:v3:os-agents": "rule:admin_api",
    "compute_extension:attach_interfaces": "",
    "compute_extension:v3:os-attach-interfaces": "",
    "compute_extension:baremetal_nodes": "rule:admin_api",
    "compute_extension:v3:os-baremetal-nodes": "rule:admin_api",
    "compute_extension:cells": "rule:admin_api",
    "compute_extension:v3:os-cells": "rule:admin_api",
    "compute_extension:certificates": "",
    "compute_extension:v3:os-certificates": "",
    "compute_extension:cloudpipe": "rule:admin_api",
    "compute_extension:cloudpipe_update": "rule:admin_api",
    "compute_extension:console_output": "",
    "compute_extension:v3:consoles:discoverable": "",
    "compute_extension:v3:os-console-output": "",
    "compute_extension:consoles": "",
    "compute_extension:v3:os-remote-consoles": "",
    "compute_extension:coverage_ext": "rule:admin_api",
    "compute_extension:v3:os-coverage": "rule:admin_api",
    "compute_extension:createserverext": "",
    "compute_extension:deferred_delete": "",
    "compute_extension:v3:os-deferred-delete": "",
    "compute_extension:disk_config": "",
    "compute_extension:evacuate": "rule:admin_api",
    "compute_extension:v3:os-evacuate": "rule:admin_api",
    "compute_extension:extended_server_attributes": "rule:admin_api",
    "compute_extension:v3:os-extended-server-attributes": "rule:admin_api",
    "compute_extension:extended_status": "",
    "compute_extension:v3:os-extended-status": "",
    "compute_extension:extended_availability_zone": "",
    "compute_extension:v3:os-extended-availability-zone": "",
    "compute_extension:extended_ips": "",
    "compute_extension:extended_ips_mac": "",
    "compute_extension:extended_vif_net": "",
    "compute_extension:v3:extension_info:discoverable": "",
    "compute_extension:extended_volumes": "",
    "compute_extension:v3:os-extended-volumes": "",
    "compute_extension:v3:os-extended-volumes:attach": "",
    "compute_extension:v3:os-extended-volumes:detach": "",
    "compute_extension:fixed_ips": "rule:admin_api",
    "compute_extension:v3:os-fixed-ips:discoverable": "",
    "compute_extension:v3:os-fixed-ips": "rule:admin_api",
    "compute_extension:flavor_access": "",
    "compute_extension:v3:os-flavor-access": "",
    "compute_extension:flavor_disabled": "",
    "compute_extension:v3:os-flavor-disabled": "",
    "compute_extension:flavor_rxtx": "",
    "compute_extension:v3:os-flavor-rxtx": "",
    "compute_extension:flavor_swap": "",
    "compute_extension:flavorextradata": "",
    "compute_extension:flavorextraspecs:index": "",
    "compute_extension:flavorextraspecs:show": "",
```

```
    "compute_extension:flavorextraspecs:create": "rule:admin_api",
    "compute_extension:flavorextraspecs:update": "rule:admin_api",
    "compute_extension:flavorextraspecs:delete": "rule:admin_api",
    "compute_extension:v3:flavor-extra-specs:index": "",
    "compute_extension:v3:flavor-extra-specs:show": "",
    "compute_extension:v3:flavor-extra-specs:create": "rule:admin_api",
    "compute_extension:v3:flavor-extra-specs:update": "rule:admin_api",
    "compute_extension:v3:flavor-extra-specs:delete": "rule:admin_api",
    "compute_extension:flavormanage": "rule:admin_api",
    "compute_extension:floating_ip_dns": "",
    "compute_extension:floating_ip_pools": "",
    "compute_extension:floating_ips": "",
    "compute_extension:floating_ips_bulk": "rule:admin_api",
    "compute_extension:fping": "",
    "compute_extension:fping:all_tenants": "rule:admin_api",
    "compute_extension:hide_server_addresses": "is_admin:False",
    "compute_extension:v3:os-hide-server-addresses": "is_admin:False",
    "compute_extension:hosts": "rule:admin_api",
    "compute_extension:v3:os-hosts": "rule:admin_api",
    "compute_extension:hypervisors": "rule:admin_api",
    "compute_extension:v3:os-hypervisors": "rule:admin_api",
    "compute_extension:image_size": "",
    "compute_extension:v3:os-image-metadata": "",
    "compute_extension:v3:os-images": "",
    "compute_extension:instance_actions": "",
    "compute_extension:v3:os-instance-actions": "",
    "compute_extension:instance_actions:events": "rule:admin_api",
    "compute_extension:v3:os-instance-actions:events": "rule:admin_api",
    "compute_extension:instance_usage_audit_log": "rule:admin_api",
    "compute_extension:v3:os-instance-usage-audit-log": "rule:admin_api",
    "compute_extension:v3:ips:discoverable": "",
    "compute_extension:keypairs": "",
    "compute_extension:keypairs:index": "",
    "compute_extension:keypairs:show": "",
    "compute_extension:keypairs:create": "",
    "compute_extension:keypairs:delete": "",
    "compute_extension:v3:os-keypairs:discoverable": "",
    "compute_extension:v3:os-keypairs": "",
    "compute_extension:v3:os-keypairs:index": "",
    "compute_extension:v3:os-keypairs:show": "",
    "compute_extension:v3:os-keypairs:create": "",
    "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": ""
}
```

Dashboard log files

The dashboard is served to users through the Apache web server (`httpd`).

As a result, dashboard-related logs appear in files in the `/var/log/httpd` or `/var/log/apache2` directory on the system where the dashboard is hosted. The following table describes these files:

Table 4.1. Dashboard/httpd log files

Log file	Description
`access_log`	Logs all attempts to access the web server.
`error_log`	Logs all unsuccessful attempts to access the web server, along with the reason that each attempt failed.

5. Database service

Table of Contents

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 5.1. Description of API configuration options

Configuration option = Default value	Description
[DEFAULT]	
`admin_roles` = *admin*	(ListOpt) Roles to add to an admin user.
`api_paste_config` = *api-paste.ini*	(StrOpt) File name for the paste.deploy config for trove-api.
`bind_host` = *0.0.0.0*	(StrOpt) IP address the API server will listen on.
`bind_port` = *8779*	(IntOpt) Port the API server will listen on.
`black_list_regex` = *None*	(StrOpt) Exclude IP addresses that match this regular expression.
`db_api_implementation` = *trove.db.sqlalchemy.api*	(StrOpt) API Implementation for Trove database access.
`hostname_require_valid_ip` = *True*	(BoolOpt) Require user hostnames to be valid IP addresses.
`http_delete_rate` = *200*	(IntOpt) Maximum number of HTTP 'DELETE' requests (per minute).
`http_get_rate` = *200*	(IntOpt) Maximum number of HTTP 'GET' requests (per minute).
`http_mgmt_post_rate` = *200*	(IntOpt) Maximum number of management HTTP 'POST' requests (per minute).
`http_post_rate` = *200*	(IntOpt) Maximum number of HTTP 'POST' requests (per minute).
`http_put_rate` = *200*	(IntOpt) Maximum number of HTTP 'PUT' requests (per minute).
`injected_config_location` = */etc/trove/conf.d*	(StrOpt) Path to folder on the Guest where config files will be injected during instance creation.
`instances_page_size` = *20*	(IntOpt) Page size for listing instances.
`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).
`os_region_name` = *RegionOne*	(StrOpt) Region name of this node. Used when searching catalog.
`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.

Configuration option = Default value	Description
trove_api_workers = *None*	(IntOpt) Number of workers for the API service. The default will be the number of CPUs available.
trove_auth_url = *http://0.0.0.0:5000/v2.0*	(StrOpt) Trove authentication URL.
trove_conductor_workers = *None*	(IntOpt) Number of workers for the Conductor service. The default will be the number of CPUs available.
trove_security_group_name_prefix = *SecGroup*	(StrOpt) Prefix to use when creating Security Groups.
trove_security_group_rule_cidr = *0.0.0.0/0*	(StrOpt) CIDR to use when creating Security Group Rules.
trove_security_groups_support = *True*	(BoolOpt) Whether Trove should add Security Groups on create.
users_page_size = *20*	(IntOpt) Page size for listing users.

Table 5.2. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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 un-

Configuration option = Default value	Description
	known 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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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.

Configuration option = Default value	Description
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 5.3. Description of backup configuration options

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 (in bytes) to stream to the 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_runner = *trove.guestagent.backup.backup_types.InnoBackupEx*	(StrOpt) Runner to use for backups.
backup_runner_options = *{}*	(DictOpt) Additional options to be passed to the backup runner.
backup_segment_max_size = *2147483648*	(IntOpt) Maximum size (in bytes) of each segment of the backup file.
backup_swift_container = *database_backups*	(StrOpt) Swift container to put backups in.
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) Page size for listing backups.

Table 5.4. Description of CA and SSL configuration options

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 5.5. Description of clients configuration options

Configuration option = Default value	Description
[DEFAULT]	
remote_cinder_client = *trove.common.remote.cinder_client*	(StrOpt) Client to send Cinder calls to.
remote_dns_client = *trove.common.remote.dns_client*	(StrOpt) Client to send DNS calls to.
remote_guest_client = *trove.common.remote.guest_client*	(StrOpt) Client to send Guest Agent calls to.
remote_heat_client = *trove.common.remote.heat_client*	(StrOpt) Client to send Heat calls to.

Configuration option = Default value	Description
remote_neutron_client = *trove.common.remote.neutron_client*	(StrOpt) Client to send Neutron calls to.
remote_nova_client = *trove.common.remote.nova_client*	(StrOpt) Client to send Nova calls to.
remote_swift_client = *trove.common.remote.swift_client*	(StrOpt) Client to send Swift calls to.

Table 5.6. Description of cluster configuration options

Configuration option = Default value	Description
[DEFAULT]	
cluster_delete_time_out = *180*	(IntOpt) Maximum time (in seconds) to wait for a cluster delete.
cluster_usage_timeout = *675*	(IntOpt) Maximum time (in seconds) to wait for a cluster to become active.
clusters_page_size = *20*	(IntOpt) Page size for listing clusters.

Table 5.7. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
configurations_page_size = *20*	(IntOpt) Page size for listing configurations.
databases_page_size = *20*	(IntOpt) Page size for listing databases.
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.
default_neutron_networks =	(ListOpt) List of IDs for management networks which should be attached to the instance regardless of what NICs are specified in the create API call.
default_password_length = *36*	(IntOpt) Character length of generated passwords.
expected_filetype_suffixes = *json*	(ListOpt) Filetype endings not to be reattached to an ID by the utils method correct_id_with_req.
host = *0.0.0.0*	(StrOpt) Host to listen for RPC messages.
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
pybasedir = */usr/lib/python/site-packages/trove/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) Message queue name the Taskmanager will listen to.
template_path = */etc/trove/templates/*	(StrOpt) Path which leads to datastore templates.
usage_timeout = *600*	(IntOpt) Maximum time (in seconds) to wait for a Guest to become active.
[keystone_authtoken]	
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.

Table 5.8. Description of Compute configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
ip_regex = *None*	(StrOpt) List IP addresses that match this regular expression.
nova_compute_endpoint_type = *publicURL*	(StrOpt) Service endpoint type to use when searching catalog.
nova_compute_service_type = *compute*	(StrOpt) Service type to use when searching catalog.
nova_compute_url = *None*	(StrOpt) URL without the tenant segment.
root_grant = *ALL*	(ListOpt) Permissions to grant to the 'root' user.
root_grant_option = *True*	(BoolOpt) Assign the 'root' user GRANT permissions.

Table 5.9. Description of logging configuration options

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
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).
[profiler]	
enabled = *False*	(BoolOpt) If False fully disable profiling feature.
trace_sqlalchemy = *True*	(BoolOpt) If False doesn't trace SQL requests.

Table 5.10. Description of DNS configuration options

Configuration option = Default value	Description
[DEFAULT]	
dns_account_id =	(StrOpt) Tenant ID for DNSaaS.
dns_auth_url =	(StrOpt) Authentication URL for DNSaaS.
dns_domain_id =	(StrOpt) Domain ID used for adding DNS entries.
dns_domain_name =	(StrOpt) Domain name used for adding DNS entries.
dns_driver = *trove.dns.driver.DnsDriver*	(StrOpt) Driver for DNSaaS.
dns_endpoint_url = *0.0.0.0*	(StrOpt) Endpoint URL for DNSaaS.
dns_hostname =	(StrOpt) Hostname used for adding DNS entries.
dns_instance_entry_factory = *trove.dns.driver.DnsInstanceEntryFactory*	(StrOpt) Factory for adding DNS entries.
dns_management_base_url =	(StrOpt) Management URL for DNSaaS.
dns_passkey =	(StrOpt) Passkey for DNSaaS.
dns_region =	(StrOpt) Region name for DNSaaS.
dns_service_type =	(StrOpt) Service Type for DNSaaS.

Configuration option = Default value	Description
dns_time_out = *120*	(IntOpt) Maximum time (in seconds) to wait for a DNS entry add.
dns_ttl = *300*	(IntOpt) Time (in seconds) before a refresh of DNS information occurs.
dns_username =	(StrOpt) Username for DNSaaS.
trove_dns_support = *False*	(BoolOpt) Whether Trove should add DNS entries on create (using Designate DNSaaS).

Table 5.11. Description of guest agent configuration options

Configuration option = Default value	Description
[DEFAULT]	
agent_call_high_timeout = *60*	(IntOpt) Maximum time (in seconds) to wait for Guest Agent 'slow' requests (such as restarting the database).
agent_call_low_timeout = *5*	(IntOpt) Maximum time (in seconds) to wait for Guest Agent 'quick'requests (such as retrieving a list of users or databases).
agent_heartbeat_expiry = *60*	(IntOpt) Time (in seconds) after which a guest is considered unreachable
agent_heartbeat_time = *10*	(IntOpt) Maximum time (in seconds) for the Guest Agent to reply to a heartbeat request.
agent_replication_snapshot_timeout = *36000*	(IntOpt) Maximum time (in seconds) to wait for taking a Guest Agent replication snapshot.
guest_config = */etc/trove/trove-guestagent.conf*	(StrOpt) Path to the Guest Agent config file to be injected during instance creation.
guest_id = *None*	(StrOpt) ID of the Guest Instance.
guest_info = *guest_info.conf*	(StrOpt) The guest info filename found in the injected config location. If a full path is specified then it will be used as the path to the guest info file
ignore_dbs = *lost+found, #mysql50#lost+found, mysql, information_schema*	(ListOpt) Databases to exclude when listing databases.
ignore_users = *os_admin, root*	(ListOpt) Users to exclude when listing users.
mount_options = *defaults,noatime*	(StrOpt) Options to use when mounting a volume.
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 for an active Guest.

Table 5.12. Description of Orchestration module configuration options

Configuration option = Default value	Description
[DEFAULT]	
heat_endpoint_type = *publicURL*	(StrOpt) Service endpoint type to use when searching catalog.
heat_service_type = *orchestration*	(StrOpt) Service type to use when searching catalog.
heat_time_out = *60*	(IntOpt) Maximum time (in seconds) to wait for a Heat request to complete.
heat_url = *None*	(StrOpt) URL without the tenant segment.

Table 5.13. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
format_options = *-m 5*	(StrOpt) Options to use when formatting a volume.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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) Regular expression to match Trove network labels.
publish_errors = *False*	(BoolOpt) Enables or disables publication of 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 will change in J to honor RFC5424.

Configuration option = Default value	Description
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 5.14. Description of network configuration options

Configuration option = Default value	Description
[DEFAULT]	
network_driver = *trove.network.nova.NovaNetwork*	(StrOpt) Describes the actual network manager used for the management of network attributes (security groups, floating IPs, etc.).
neutron_endpoint_type = *publicURL*	(StrOpt) Service endpoint type to use when searching catalog.
neutron_service_type = *network*	(StrOpt) Service type to use when searching catalog.
neutron_url = *None*	(StrOpt) URL without the tenant segment.

Table 5.15. Description of nova configuration options

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 5.16. Description of quota configuration options

Configuration option = Default value	Description
[DEFAULT]	
max_accepted_volume_size = *5*	(IntOpt) Default maximum volume size (in GB) 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 5.17. Description of Redis configuration options

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 5.18. Description of swift configuration options

Configuration option = Default value	Description
[DEFAULT]	
swift_endpoint_type = *publicURL*	(StrOpt) Service endpoint type to use when searching catalog.
swift_service_type = *object-store*	(StrOpt) Service type to use when searching catalog.
swift_url = *None*	(StrOpt) URL ending in AUTH_.

Table 5.19. Description of taskmanager configuration options

Configuration option = Default value	Description
[DEFAULT]	
cloudinit_location = */etc/trove/cloudinit*	(StrOpt) Path to folder with cloudinit scripts.
datastore_manager = *None*	(StrOpt) Manager class in the Guest Agent, set up by the Taskmanager on instance provision.
datastore_registry_ext = *{}*	(DictOpt) Extension for default datastore managers. Allows the use of custom managers for each of the datastores supported by 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) Maximum time (in seconds) to wait for a server reboot.
resize_time_out = *600*	(IntOpt) Maximum time (in seconds) to wait for a server resize.
restore_usage_timeout = *36000*	(IntOpt) Maximum time (in seconds) to wait for a Guest instance restored from a backup to become active.
revert_time_out = *600*	(IntOpt) Maximum time (in seconds) to wait for a server resize revert.
server_delete_time_out = *60*	(IntOpt) Maximum time (in seconds) to wait for a server delete.
state_change_wait_time = *180*	(IntOpt) Maximum time (in seconds) to wait for a state change.
update_status_on_fail = *True*	(BoolOpt) Set the service and instance task statuses to ERROR when an instance fails to become active within the configured usage_timeout.
usage_sleep_time = *5*	(IntOpt) Time to sleep during the check for an active Guest.
use_heat = *False*	(BoolOpt) Use Heat for provisioning.
use_nova_server_config_drive = *False*	(BoolOpt) Use config drive for file injection when booting instance.
use_nova_server_volume = *False*	(BoolOpt) Whether to provision a Cinder volume for the Nova instance.
verify_swift_checksum_on_restore = *True*	(BoolOpt) Enable verification of Swift checksum before starting restore. Makes sure the checksum of original backup matches the checksum of the Swift backup file.

Table 5.20. Description of upgrades configuration options

Configuration option = Default value	Description
[upgrade_levels]	
conductor = *icehouse*	(StrOpt) Set a version cap for messages sent to conductor services

Configuration option = Default value	Description
guestagent = *icehouse*	(StrOpt) Set a version cap for messages sent to guestagent services
taskmanager = *icehouse*	(StrOpt) Set a version cap for messages sent to taskmanager services

Table 5.21. Description of volume configuration options

Configuration option = Default value	Description
[DEFAULT]	
block_device_mapping = *vdb*	(StrOpt) Block device to map onto the created instance.
cinder_endpoint_type = *publicURL*	(StrOpt) Service endpoint type to use when searching catalog.
cinder_service_type = *volumev2*	(StrOpt) Service type to use when searching catalog.
cinder_url = *None*	(StrOpt) URL without the tenant segment.
cinder_volume_type = *None*	(StrOpt) Volume type to use when provisioning a Cinder volume.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
trove_volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.
volume_format_timeout = *120*	(IntOpt) Maximum time (in seconds) to wait for a volume format.
volume_fstype = *ext3*	(StrOpt) File system type used to format a volume.
volume_time_out = *60*	(IntOpt) Maximum time (in seconds) to wait for a volume attach.

Configure the database

Use the options to configure the used databases:

Table 5.22. Description of database configuration options

Configuration option = Default value	Description
[DEFAULT]	
sql_query_logging = *False*	(BoolOpt) Allow insecure logging while executing queries through SQLAlchemy.
[database]	
connection = *sqlite:///trove_test.sqlite*	(StrOpt) SQL Connection.
idle_timeout = *3600*	(IntOpt) No help text available for this option.
query_log = *False*	(BoolOpt) No help text available for this option.

Table 5.23. Description of Cassandra database configuration options

Configuration option = Default value	Description
[cassandra]	
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = *None*	(StrOpt) Namespace to load backup strategies from.

Configuration option = Default value	Description
backup_strategy = *None*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
mount_point = */var/lib/cassandra*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_strategy = *None*	(StrOpt) Default strategy for replication.
restore_namespace = *None*	(StrOpt) Namespace to load restore strategies from.
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).
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.24. Description of Couchbase database configuration options

Configuration option = Default value	Description
[couchbase]	
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on 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.experimental.couchbase_impl*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *CbBackup*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
mount_point = */var/lib/couchbase*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_strategy = *None*	(StrOpt) Default strategy for replication.
restore_namespace = *trove.guestagent.strategies.restore.experimental.couchbase_impl*	(StrOpt) Namespace to load restore strategies from.
root_on_create = *True*	(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 = *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).
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.
[couchdb]	
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = *None*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *None*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.

Configuration option = Default value	Description
mount_point = /var/lib/couchdb	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_strategy = None	(StrOpt) Default strategy for replication.
restore_namespace = None	(StrOpt) Namespace to load restore strategies from.
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 = 5984	(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).
volume_support = True	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.25. Description of DB2 database configuration options

Configuration option = Default value	Description
[db2]	
backup_incremental_strategy = {}	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = None	(StrOpt) Namespace to load backup strategies from.
backup_strategy = None	(StrOpt) Default strategy to perform backups.
device_path = /dev/vdb	(StrOpt) Device path for volume if volume support is enabled.
ignore_users = PUBLIC, DB2INST1	(ListOpt) No help text available for this option.
mount_point = /home/db2inst1/db2inst1	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_strategy = None	(StrOpt) Default strategy for replication.
restore_namespace = None	(StrOpt) Namespace to load restore strategies from.
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 = 50000	(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).
volume_support = True	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.26. Description of MongoDB database configuration options

Configuration option = Default value	Description
[mongodb]	
api_strategy = trove.common.strategies.cluster.experimental.mongodb.api.MongoDbAPIStrategy	(StrOpt) Class that implements datastore-specific API logic.

Configuration option = Default value	Description
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = *None*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *None*	(StrOpt) Default strategy to perform backups.
cluster_support = *True*	(BoolOpt) Enable clusters to be created and managed.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
guestagent_strategy = *trove.common.strategies.cluster.experimental.mongodb.guestagent.MongoDbGuestAgentStrategy*	(StrOpt) Class that implements datastore-specific Guest Agent API logic
mount_point = */var/lib/mongodb*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
num_config_servers_per_cluster = *3*	(IntOpt) The number of config servers to create per cluster.
num_query_routers_per_cluster = *1*	(IntOpt) The number of query routers (mongos) to create per cluster.
replication_strategy = *None*	(StrOpt) Default strategy for replication.
restore_namespace = *None*	(StrOpt) Namespace to load restore strategies from.
taskmanager_strategy = *trove.common.strategies.cluster.experimental.mongodb.taskmanager.MongoDbTaskManagerStrategy*	(StrOpt) Class that implements datastore-specific task manager logic
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).
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.27. Description of MySQL database configuration options

Configuration option = Default value	Description
[mysql]	
backup_incremental_strategy = *{'InnoBackupEx': 'InnoBackupExIncremental'}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental backup, the runner will use the default full backup.
backup_namespace = *trove.guestagent.strategies.backup.mysql_impl*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *InnoBackupEx*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
mount_point = */var/lib/mysql*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_namespace = *trove.guestagent.strategies.replication.mysql_gtid*	(StrOpt) Namespace to load replication strategies from.
replication_strategy = *MysqlGTIDReplication*	(StrOpt) Default strategy for replication.
restore_namespace = *trove.guestagent.strategies.restore.mysql_impl*	(StrOpt) Namespace to load restore strategies from.
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.

Configuration option = Default value	Description
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) Maximum time (in seconds) to wait for a Guest to become active.
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.28. Description of Percona database configuration options

Configuration option = Default value	Description
[percona]	
backup_incremental_strategy = *{'InnoBackupEx': 'InnoBackupExIncremental'}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental backup, the runner will use the default full backup.
backup_namespace = *trove.guestagent.strategies.backup.mysql_impl*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *InnoBackupEx*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
mount_point = */var/lib/mysql*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_namespace = *trove.guestagent.strategies.replication.mysql_gtid*	(StrOpt) Namespace to load replication strategies from.
replication_password = *NETOU7897NNLOU*	(StrOpt) Password for replication slave user.
replication_strategy = *MysqlGTIDReplication*	(StrOpt) Default strategy for replication.
replication_user = *slave_user*	(StrOpt) Userid for replication slave.
restore_namespace = *trove.guestagent.strategies.restore.mysql_impl*	(StrOpt) Namespace to load restore strategies from.
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) Maximum time (in seconds) to wait for a Guest to become active.
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.29. Description of PostgreSQL database configuration options

Configuration option = Default value	Description
[postgresql]	

Configuration option = Default value	Description
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on 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.experimental.postgresql_impl*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *PgDump*	(StrOpt) Default strategy to perform backups.
device_path = */dev/vdb*	(StrOpt) No help text available for this option.
ignore_dbs = *postgres*	(ListOpt) No help text available for this option.
ignore_users = *os_admin, postgres, root*	(ListOpt) No help text available for this option.
mount_point = */var/lib/postgresql*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
restore_namespace = *trove.guestagent.strategies.restore.experimental.postgresql_impl*	(StrOpt) Namespace to load restore strategies from.
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 = *5432*	(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).
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.30. Description of Redis database configuration options

Configuration option = Default value	Description
[redis]	
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = *None*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *None*	(StrOpt) Default strategy to perform backups.
device_path = *None*	(StrOpt) Device path for volume if volume support is enabled.
mount_point = */var/lib/redis*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
replication_strategy = *None*	(StrOpt) Default strategy for replication.
restore_namespace = *None*	(StrOpt) Namespace to load restore strategies from.
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).
volume_support = *False*	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.31. Description of Vertica database configuration options

Configuration option = Default value	Description
[vertica]	
api_strategy = *trove.common.strategies.cluster.experimental.vertica.api.VerticaAPIStrategy*	(StrOpt) Class that implements datastore-specific API logic.
backup_incremental_strategy = *{}*	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
backup_namespace = *None*	(StrOpt) Namespace to load backup strategies from.
backup_strategy = *None*	(StrOpt) Default strategy to perform backups.
cluster_member_count = *3*	(IntOpt) Number of members in Vertica cluster.
cluster_support = *True*	(BoolOpt) Enable clusters to be created and managed.
device_path = */dev/vdb*	(StrOpt) Device path for volume if volume support is enabled.
guestagent_strategy = *trove.common.strategies.cluster.experimental.vertica.guestagent.VerticaGuestAgentStrategy*	(StrOpt) Class that implements datastore-specific Guest Agent API logic.
mount_point = */var/lib/vertica*	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
readahead_size = *2048*	(IntOpt) Size(MB) to be set as readahead_size for data volume
replication_strategy = *None*	(StrOpt) Default strategy for replication.
restore_namespace = *None*	(StrOpt) Namespace to load restore strategies from.
taskmanager_strategy = *trove.common.strategies.cluster.experimental.vertica.taskmanager.VerticaTaskManagerStrategy*	(StrOpt) Class that implements datastore-specific task manager logic.
tcp_ports = *5433, 5434, 22, 5444, 5450, 4803*	(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 = *5433, 4803, 4804, 6453*	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True).
volume_support = *True*	(BoolOpt) Whether to provision a Cinder volume for datadir.

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 5.32. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.

Configuration option = Default value	Description
fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
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_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

Configure Qpid

Use these options to configure the Qpid messaging system:

Table 5.33. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
qpid_hostname = localhost	(StrOpt) Qpid broker hostname.

Configuration option = Default value	Description
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_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Configure ZeroMq

Use these options to configure the ZeroMq messaging system:

Table 5.34. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 5.35. Description of AMQP configuration options

Configuration option = Default value	Description
[DEFAULT]	
conductor_manager = *trove.conductor.manager.Manager*	(StrOpt) Qualified class name to use for conductor manager.
conductor_queue = *trove-conductor*	(StrOpt) Message queue name the Conductor will listen on.

Configuration option = Default value	Description
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.
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_service_id = *{'vertica': 'a8d805ae-a3b2-c4fd-gb23-b62cee5201ae', 'db2': 'e040cd37-263d-4869-aaa6-c62aa97523b5', 'postgresql': 'ac277e0d-4f21-40aa-b347-1ea31e571720', 'mysql': '2f3ff068-2bfb-4f70-9a9d-a6bb65bc084b', 'couchbase': 'fa62fe68-74d9-4779-a24e-36f19602c415', 'mongodb': 'c8c907af-7375-456f-b929-b637ff9209ee', 'couchdb': 'f0a9ab7b-66f7-4352-93d7-071521d44c7c', 'redis': 'b216ffc5-1947-456c-a4cf-70f94c05f7d0', 'cassandra': '459a230d-4e97-4344-9067-2a54a310b0ed'}*	(DictOpt) Unique ID to tag notification events.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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 5.36. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
matchmaker_heartbeat_freq = *300*	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = *600*	(IntOpt) Heartbeat time-to-live.
num_tries = *3*	(IntOpt) Number of times to check if a volume exists.
report_interval = *10*	(IntOpt) The interval (in seconds) which periodic tasks are run.
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate

Configuration option = Default value	Description
`ssl_key_password` = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
`trace` = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

New, updated and deprecated options in Kilo for Database service

Table 5.37. New options

Option = default value	(Type) Help string
[DEFAULT] agent_heartbeat_expiry = 60	(IntOpt) Time (in seconds) after which a guest is considered unreachable
[DEFAULT] cinder_endpoint_type = publicURL	(StrOpt) Service endpoint type to use when searching catalog.
[DEFAULT] guest_info = guest_info.conf	(StrOpt) The guest info filename found in the injected config location. If a full path is specified then it will be used as the path to the guest info file
[DEFAULT] heat_endpoint_type = publicURL	(StrOpt) Service endpoint type to use when searching catalog.
[DEFAULT] injected_config_location = /etc/trove/conf.d	(StrOpt) Path to folder on the Guest where config files will be injected during instance creation.
[DEFAULT] neutron_endpoint_type = publicURL	(StrOpt) Service endpoint type to use when searching catalog.
[DEFAULT] nova_compute_endpoint_type = publicURL	(StrOpt) Service endpoint type to use when searching catalog.
[DEFAULT] swift_endpoint_type = publicURL	(StrOpt) Service endpoint type to use when searching catalog.
[DEFAULT] 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.
[couchdb] backup_incremental_strategy = {}	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
[couchdb] backup_namespace = None	(StrOpt) Namespace to load backup strategies from.
[couchdb] backup_strategy = None	(StrOpt) Default strategy to perform backups.
[couchdb] device_path = /dev/vdb	(StrOpt) Device path for volume if volume support is enabled.
[couchdb] mount_point = /var/lib/couchdb	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
[couchdb] replication_strategy = None	(StrOpt) Default strategy for replication.
[couchdb] restore_namespace = None	(StrOpt) Namespace to load restore strategies from.
[couchdb] 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.
[couchdb] tcp_ports = 5984	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True).
[couchdb] 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).

Option = default value	(Type) Help string
[couchdb] volume_support = True	(BoolOpt) Whether to provision a Cinder volume for datadir.
[database] connection = sqlite:///trove_test.sqlite	(StrOpt) SQL Connection.
[database] idle_timeout = 3600	(IntOpt) None
[database] query_log = False	(BoolOpt) None
[db2] backup_incremental_strategy = {}	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
[db2] backup_namespace = None	(StrOpt) Namespace to load backup strategies from.
[db2] backup_strategy = None	(StrOpt) Default strategy to perform backups.
[db2] device_path = /dev/vdb	(StrOpt) Device path for volume if volume support is enabled.
[db2] ignore_users = PUBLIC, DB2INST1	(ListOpt) None
[db2] mount_point = /home/db2inst1/db2inst1	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
[db2] replication_strategy = None	(StrOpt) Default strategy for replication.
[db2] restore_namespace = None	(StrOpt) Namespace to load restore strategies from.
[db2] 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.
[db2] tcp_ports = 50000	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True).
[db2] 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).
[db2] volume_support = True	(BoolOpt) Whether to provision a Cinder volume for datadir.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.

Option = default value	(Type) Help string
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[profiler] enabled = False	(BoolOpt) If False fully disable profiling feature.
[profiler] trace_sqlalchemy = True	(BoolOpt) If False doesn't trace SQL requests.
[upgrade_levels] conductor = icehouse	(StrOpt) Set a version cap for messages sent to conductor services
[upgrade_levels] guestagent = icehouse	(StrOpt) Set a version cap for messages sent to guestagent services
[upgrade_levels] taskmanager = icehouse	(StrOpt) Set a version cap for messages sent to taskmanager services
[vertica] api_strategy = trove.common.strategies.cluster.experimental.vertica.api.VerticaAPIStrategy	(StrOpt) Class that implements datastore-specific API logic.
[vertica] backup_incremental_strategy = {}	(DictOpt) Incremental Backup Runner based on the default strategy. For strategies that do not implement an incremental, the runner will use the default full backup.
[vertica] backup_namespace = None	(StrOpt) Namespace to load backup strategies from.
[vertica] backup_strategy = None	(StrOpt) Default strategy to perform backups.
[vertica] cluster_member_count = 3	(IntOpt) Number of members in Vertica cluster.
[vertica] cluster_support = True	(BoolOpt) Enable clusters to be created and managed.
[vertica] device_path = /dev/vdb	(StrOpt) Device path for volume if volume support is enabled.
[vertica] guestagent_strategy = trove.common.strategies.cluster.experimental.vertica.guestagent.VerticaGuestAgentStrategy	(StrOpt) Class that implements datastore-specific Guest Agent API logic.
[vertica] mount_point = /var/lib/vertica	(StrOpt) Filesystem path for mounting volumes if volume support is enabled.
[vertica] readahead_size = 2048	(IntOpt) Size(MB) to be set as readahead_size for data volume
[vertica] replication_strategy = None	(StrOpt) Default strategy for replication.
[vertica] restore_namespace = None	(StrOpt) Namespace to load restore strategies from.
[vertica] taskmanager_strategy = trove.common.strategies.cluster.experimental.vertica.taskmanager.VerticaTaskManagerStrategy	(StrOpt) Class that implements datastore-specific task manager logic.
[vertica] tcp_ports = 5433, 5434, 22, 5444, 5450, 4803	(ListOpt) List of TCP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True).
[vertica] udp_ports = 5433, 4803, 4804, 6453	(ListOpt) List of UDP ports and/or port ranges to open in the security group (only applicable if trove_security_groups_support is True).
[vertica] volume_support = True	(BoolOpt) Whether to provision a Cinder volume for datadir.

Table 5.38. New default values

Option	Previous default value	New default value
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO,	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO,

Option	Previous default value	New default value
	oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN	oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] guest_config	$pybasedir/etc/trove/trove-guestagent.conf.sample	/etc/trove/trove-guestagent.conf
[DEFAULT] ignore_dbs	lost+found, mysql, information_schema	lost+found, #mysql50#lost+found, mysql, information_schema
[DEFAULT] notification_service_id	{'postgresql': 'ac277e0d-4f21-40aa-b347-1ea31e571720', 'couchbase': 'fa62fe68-74d9-4779-a24e-36f19602c415', 'mongodb': 'c8c907af-7375-456f-b929-b637ff9209ee', 'redis': 'b216ffc5-1947-456c-a4cf-70f94c05f7d0', 'mysql': '2f3ff068-2bfb-4f70-9a9d-a6bb65bc084b', 'cassandra': '459a230d-4e97-4344-9067-2a54a310b0ed'}	{'vertica': 'a8d805ae-a3b2-c4fd-gb23-b62cee5201ae', 'db2': 'e040cd37-263d-4869-aaa6-c62aa97523b5', 'postgresql': 'ac277e0d-4f21-40aa-b347-1ea31e571720', 'mysql': '2f3ff068-2bfb-4f70-9a9d-a6bb65bc084b', 'couchbase': 'fa62fe68-74d9-4779-a24e-36f19602c415', 'mongodb': 'c8c907af-7375-456f-b929-b637ff9209ee', 'couchdb': 'f0a9ab7b-66f7-4352-93d7-071521d44c7c', 'redis': 'b216ffc5-1947-456c-a4cf-70f94c05f7d0', 'cassandra': '459a230d-4e97-4344-9067-2a54a310b0ed'}
[DEFAULT] os_region_name	None	RegionOne
[DEFAULT] rpc_backend	trove.openstack.common.rpc.impl_kombu	rabbit
[DEFAULT] rpc_zmq_matchmaker	trove.openstack.common.rpc.matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost
[couchbase] backup_namespace	trove.guestagent.strategies.backup.couchbase_impl	trove.guestagent.strategies.backup.experimental.couchbase_
[couchbase] restore_namespace	trove.guestagent.strategies.restore.couchbase_impl	trove.guestagent.strategies.restore.experimental.couchbase_
[mongodb] api_strategy	trove.common.strategies.mongodb.api.MongoDBAPIStrategy	trove.common.strategies.cluster.experimental.mongodb.api.
[mongodb] guestagent_strategy	trove.common.strategies.mongodb.guestagent.MongoDBGuestAgentStrategy	trove.common.strategies.cluster.experimental.mongodb.gue
[mongodb] taskmanager_strategy	trove.common.strategies.mongodb.taskmanager.MongoDBTaskManagerStrategy	trove.common.strategies.cluster.experimental.mongodb.task
[mysql] replication_namespace	trove.guestagent.strategies.replication.mysql_binlog	trove.guestagent.strategies.replication.mysql_gtid
[mysql] replication_strategy	MysqlBinlogReplication	MysqlGTIDReplication
[percona] replication_namespace	trove.guestagent.strategies.replication.mysql_binlog	trove.guestagent.strategies.replication.mysql_gtid
[percona] replication_strategy	MysqlBinlogReplication	MysqlGTIDReplication
[postgresql] backup_namespace	trove.guestagent.strategies.backup.postgresql_impl	trove.guestagent.strategies.backup.experimental.postgresql_
[postgresql] restore_namespace	trove.guestagent.strategies.restore.postgresql_impl	trove.guestagent.strategies.restore.experimental.postgresql_

Table 5.39. Deprecated options

Deprecated option	New Option
[DEFAULT] use_syslog	None
[rpc_notifier2] topics	[DEFAULT] notification_topics
[DEFAULT] sql_query_log	[database] query_log
[DEFAULT] sql_connection	[database] connection
[DEFAULT] sql_idle_timeout	[database] idle_timeout
[DEFAULT] log_format	None

6. Data processing service

Table of Contents

The Data processing service (sahara) provides a scalable data-processing stack and associated management interfaces.

The following tables provide a comprehensive list of the Data processing service configuration options.

Table 6.1. Description of AMQP configuration options

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.
`notification_driver` = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
`notification_level` = *INFO*	(StrOpt) Notification level for outgoing notifications
`notification_publisher_id` = *None*	(StrOpt) Notification publisher_id for outgoing notifications
`notification_topics` = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
`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 6.2. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
`admin_password` = *None*	(StrOpt) Service user password.
`admin_tenant_name` = *admin*	(StrOpt) Service tenant name.
`admin_token` = *None*	(StrOpt) 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.
`admin_user` = *None*	(StrOpt) Service username.
`auth_admin_prefix` =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
`auth_host` = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_plugin` = *None*	(StrOpt) Name of the plugin to load
`auth_port` = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_protocol` = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.

Configuration option = Default value	Description
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.

Configuration option = Default value	Description
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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 6.3. Description of CA and SSL configuration options

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 6.4. Description of clients configuration options

Configuration option = Default value	Description
[cinder]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to cinder.
api_version = *2*	(IntOpt) Version of the Cinder API to use.
ca_file = *None*	(StrOpt) Location of ca certificates file to use for cinder client requests.
[heat]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to heat.
ca_file = *None*	(StrOpt) Location of ca certificates file to use for heat client requests.
[keystone]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to keystone.

Configuration option = Default value	Description
ca_file = *None*	(StrOpt) Location of ca certificates file to use for keystone client requests.
[neutron]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to neutron.
ca_file = *None*	(StrOpt) Location of ca certificates file to use for neutron client requests.
[nova]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to nova.
ca_file = *None*	(StrOpt) Location of ca certificates file to use for nova client requests.
[swift]	
api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to swift.
ca_file = *None*	(StrOpt) Location of ca certificates file to use for swift client requests.

Table 6.5. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
admin_project_domain_name = *default*	(StrOpt) The name of the domain for the service project(ex. tenant).
admin_user_domain_name = *default*	(StrOpt) The name of the domain to which the admin user belongs.
api_workers = *0*	(IntOpt) Number of workers for Sahara API service (0 means all-in-one-thread configuration).
cleanup_time_for_incomplete_clusters = *0*	(IntOpt) Maximal time (in hours) for clusters allowed to be in states other than "Active", "Deleting" or "Error". If a cluster is not in "Active", "Deleting" or "Error" state and last update of it was longer than "cleanup_time_for_incomplete_clusters" hours ago then it will be deleted automatically. (0 value means that automatic clean up is disabled).
cluster_remote_threshold = *70*	(IntOpt) The same as global_remote_threshold, but for a single cluster.
compute_topology_file = *etc/sahara/compute.topology*	(StrOpt) File with nova compute topology. It should contain mapping between nova computes and racks.
disable_event_log = *False*	(BoolOpt) Disables event log feature.
enable_data_locality = *False*	(BoolOpt) Enables data locality for hadoop cluster. Also enables data locality for Swift used by hadoop. If enabled, 'compute_topology' and 'swift_topology' configuration parameters should point to OpenStack and Swift topology correspondingly.
enable_hypervisor_awareness = *True*	(BoolOpt) Enables four-level topology for data locality. Works only if corresponding plugin supports such mode.
enable_notifications = *False*	(BoolOpt) Enables sending notifications to Ceilometer
global_remote_threshold = *100*	(IntOpt) Maximum number of remote operations that will be running at the same time. Note that each remote operation requires its own process to run.
infrastructure_engine = *direct*	(StrOpt) An engine which will be used to provision infrastructure for Hadoop cluster.
job_binary_max_KB = *5120*	(IntOpt) Maximum length of job binary data in kilobytes that may be stored or retrieved in a single operation.

Configuration option = Default value	Description
job_canceling_timeout = *300*	(IntOpt) Timeout for canceling job execution (in seconds). Sahara will try to cancel job execution during this time.
job_workflow_postfix =	(StrOpt) Postfix for storing jobs in hdfs. Will be added to '/user/<hdfs user>/' path.
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).
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
min_transient_cluster_active_time = *30*	(IntOpt) Minimal "lifetime" in seconds for a transient cluster. Cluster is guaranteed to be "alive" within this time period.
node_domain = *novalocal*	(StrOpt) The suffix of the node's FQDN. In nova-network that is the dhcp_domain config parameter.
os_region_name = *None*	(StrOpt) Region name used to get services endpoints.
periodic_enable = *True*	(BoolOpt) Enable periodic tasks.
periodic_fuzzy_delay = *60*	(IntOpt) Range in seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0).
periodic_interval_max = *60*	(IntOpt) Max interval size between periodic tasks execution in seconds.
plugins = *vanilla, hdp, spark, cdh*	(ListOpt) List of plugins to be loaded. Sahara preserves the order of the list when returning it.
proxy_command =	(StrOpt) Proxy command used to connect to instances. If set, this command should open a netcat socket, that Sahara will use for SSH and HTTP connections. Use {host} and {port} to describe the destination. Other available keywords: {tenant_id}, {network_id}, {router_id}.
remote = *ssh*	(StrOpt) A method for Sahara to execute commands on VMs.
rootwrap_command = *sudo sahara-rootwrap /etc/sahara/rootwrap.conf*	(StrOpt) Rootwrap command to leverage. Use in conjunction with use_rootwrap=True
run_external_periodic_tasks = *True*	(BoolOpt) Some periodic tasks can be run in a separate process. Should we run them here?
swift_topology_file = *etc/sahara/swift.topology*	(StrOpt) File with Swift topology.It should contain mapping between Swift nodes and racks.
use_external_key_manager = *False*	(BoolOpt) Enable Sahara to use an external key manager service provided by the identity service catalog. Sahara will store all keys with the manager service.
use_floating_ips = *True*	(BoolOpt) If set to True, Sahara will use floating IPs to communicate with instances. To make sure that all instances have floating IPs assigned in Nova Network set "auto_assign_floating_ip=True" in nova.conf. If Neutron is used for networking, make sure that all Node Groups have "floating_ip_pool" parameter defined.
use_identity_api_v3 = *True*	(BoolOpt) Enables Sahara to use Keystone API v3. If that flag is disabled, per-job clusters will not be terminated automatically.
use_namespaces = *False*	(BoolOpt) Use network namespaces for communication (only valid to use in conjunction with use_neutron=True).
use_neutron = *False*	(BoolOpt) Use Neutron Networking (False indicates the use of Nova networking).
use_rootwrap = *False*	(BoolOpt) Use rootwrap facility to allow non-root users to run the sahara-all server instance and access pri-

Configuration option = Default value	Description
	vate network IPs (only valid to use in conjunction with use_namespaces=True)
[conductor]	
use_local = *True*	(BoolOpt) Perform sahara-conductor operations locally.
[keystone_authtoken]	
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.

Table 6.6. Description of database configuration options

Configuration option = Default value	Description
[DEFAULT]	
db_driver = *sahara.db*	(StrOpt) Driver to use for database access.
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify 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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.sqlite*	(StrOpt) The file name to use with SQLite.
sqlite_synchronous = *True*	(BoolOpt) If True, SQLite uses synchronous mode.

Configuration option = Default value	Description
use_db_reconnect = *False*	(BoolOpt) Enable the experimental use of database reconnect on connection lost.

Table 6.7. Description of domain configuration options

Configuration option = Default value	Description
[DEFAULT]	
proxy_user_domain_name = *None*	(StrOpt) The domain Sahara will use to create new proxy users for Swift object access.
proxy_user_role_names = *Member*	(ListOpt) A list of the role names that the proxy user should assume through trust for Swift object access.
use_domain_for_proxy_users = *False*	(BoolOpt) Enables Sahara to use a domain for creating temporary proxy users to access Swift. If this is enabled a domain must be created for Sahara to use.

Table 6.8. Description of logging configuration options

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 = *amqplib=WARN, qpid.messaging=INFO, stevedore=INFO, eventlet.wsgi.server=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, paramiko=WARN, requests=WARN, iso8601=WARN, oslo_messaging=INFO*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.

Configuration option = Default value	Description
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) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
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 will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 6.9. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
max_request_body_size = *114688*	(IntOpt) The maximum body size for each request, in bytes.

Table 6.10. Description of policy configuration options

Configuration option = Default value	Description
[oslo_policy]	
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.

Configuration option = Default value	Description
policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.json	(StrOpt) The JSON file that defines policies.

Table 6.11. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
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_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

Table 6.12. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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).

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) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
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.
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_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 6.13. Description of Redis configuration options

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 6.14. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP

Configuration option = Default value	Description
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 6.15. Description of timeouts configuration options

Configuration option = Default value	Description
[timeouts]	
await_attach_volumes = *10*	(IntOpt) Wait for attaching volumes to instances, in seconds
await_for_instances_active = *10800*	(IntOpt) Wait for instances to become active, in seconds
delete_instances_timeout = *10800*	(IntOpt) Wait for instances to be deleted, in seconds
detach_volume_timeout = *300*	(IntOpt) Timeout for detaching volumes from instance, in seconds
ips_assign_timeout = *10800*	(IntOpt) Assign IPs timeout, in seconds
volume_available_timeout = *10800*	(IntOpt) Wait for volumes to become available, in seconds
wait_until_accessible = *10800*	(IntOpt) Wait for instance accessibility, in seconds

Table 6.16. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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.

New, updated and deprecated options in Kilo for Data Processing service

Table 6.17. New options

Option = default value	(Type) Help string
[DEFAULT] admin_project_domain_name = default	(StrOpt) The name of the domain for the service project(ex. tenant).
[DEFAULT] admin_user_domain_name = default	(StrOpt) The name of the domain to which the admin user belongs.
[DEFAULT] api_workers = 0	(IntOpt) Number of workers for Sahara API service (0 means all-in-one-thread configuration).
[DEFAULT] cleanup_time_for_incomplete_clusters = 0	(IntOpt) Maximal time (in hours) for clusters allowed to be in states other than "Active", "Deleting" or "Error". If a cluster is not in "Active", "Deleting" or "Error" state and last update of it was longer than "cleanup_time_for_incomplete_clusters" hours ago then it will be deleted automatically. (0 value means that automatic clean up is disabled).
[DEFAULT] disable_event_log = False	(BoolOpt) Disables event log feature.
[DEFAULT] job_canceling_timeout = 300	(IntOpt) Timeout for canceling job execution (in seconds). Sahara will try to cancel job execution during this time.
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] 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).
[DEFAULT] proxy_command =	(StrOpt) Proxy command used to connect to instances. If set, this command should open a netcat socket, that Sahara will use for SSH and HTTP connections. Use {host} and {port} to describe the destination. Other available keywords: {tenant_id}, {network_id}, {router_id}.
[DEFAULT] rootwrap_command = sudo sahara-rootwrap /etc/sahara/rootwrap.conf	(StrOpt) Rootwrap command to leverage. Use in conjunction with use_rootwrap=True
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The for-

Option = default value	(Type) Help string
	mat without the APP-NAME is deprecated in I, and will be removed in J.
[DEFAULT] use_external_key_manager = False	(BoolOpt) Enable Sahara to use an external key manager service provided by the identity service catalog. Sahara will store all keys with the manager service.
[DEFAULT] use_rootwrap = False	(BoolOpt) Use rootwrap facility to allow non-root users to run the sahara-all server instance and access private network IPs (only valid to use in conjunction with use_namespaces=True)
[cinder] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to cinder.
[cinder] api_version = 2	(IntOpt) Version of the Cinder API to use.
[cinder] ca_file = None	(StrOpt) Location of ca certificates file to use for cinder client requests.
[heat] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to heat.
[heat] ca_file = None	(StrOpt) Location of ca certificates file to use for heat client requests.
[keystone] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to keystone.
[keystone] ca_file = None	(StrOpt) Location of ca certificates file to use for keystone client requests.
[neutron] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to neutron.
[neutron] ca_file = None	(StrOpt) Location of ca certificates file to use for neutron client requests.
[nova] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to nova.
[nova] ca_file = None	(StrOpt) Location of ca certificates file to use for nova client requests.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.

Option = default value	(Type) Help string
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.
[oslo_policy] policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
[oslo_policy] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[oslo_policy] policy_file = policy.json	(StrOpt) The JSON file that defines policies.
[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.
[swift] api_insecure = False	(BoolOpt) Allow to perform insecure SSL requests to swift.
[swift] ca_file = None	(StrOpt) Location of ca certificates file to use for swift client requests.
[timeouts] await_attach_volumes = 10	(IntOpt) Wait for attaching volumes to instances, in seconds
[timeouts] await_for_instances_active = 10800	(IntOpt) Wait for instances to become active, in seconds
[timeouts] delete_instances_timeout = 10800	(IntOpt) Wait for instances to be deleted, in seconds
[timeouts] detach_volume_timeout = 300	(IntOpt) Timeout for detaching volumes from instance, in seconds
[timeouts] ips_assign_timeout = 10800	(IntOpt) Assign IPs timeout, in seconds
[timeouts] volume_available_timeout = 10800	(IntOpt) Wait for volumes to become available, in seconds
[timeouts] wait_until_accessible = 10800	(IntOpt) Wait for instance accessibility, in seconds

Table 6.18. New default values

Option	Previous default value	New default value
[DEFAULT] default_log_levels	amqplib=WARN, qpid.messaging=INFO, stevedore=INFO, eventlet.wsgi.server=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, paramiko=WARN, requests=WARN, iso8601=WARN	amqplib=WARN, qpid.messaging=INFO, stevedore=INFO, eventlet.wsgi.server=WARN, sqlalchemy=WARN, boto=WARN, suds=INFO, keystone=INFO, paramiko=WARN, requests=WARN, iso8601=WARN, oslo_messaging=INFO
[DEFAULT] plugins	vanilla, hdp, spark	vanilla, hdp, spark, cdh
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost

Table 6.19. Deprecated options

Deprecated option	New Option
[DEFAULT] log-format	None
[DEFAULT] use-syslog	None
[DEFAULT] detach_volume_timeout	[timeouts] detach_volume_timeout
[DEFAULT] log_format	None
[DEFAULT] use_syslog	None

7. Identity service

Table of Contents

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.

Caching layer

Identity supports a caching layer that is above the configurable subsystems, such as token or assignment. The majority of the caching configuration options are set in the `[cache]` section. However, each section that has the capability to be cached usually has a `caching` option that will toggle caching for that specific section. By default, caching is globally disabled. Options are as follows:

Table 7.1. Description of cache configuration options

Configuration option = Default value	Description
[cache]	
`backend` = *`keystone.common.cache.noop`*	(StrOpt) Dogpile.cache backend module. It is recommended that Memcache with pooling (keystone.cache.memcache_pool) 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.
`memcache_dead_retry` = *`300`*	(IntOpt) Number of seconds memcached server is considered dead before it is tried

Configuration option = Default value	Description
	again. (dogpile.cache.memcache and keystone.cache.memcache_pool backends only).
memcache_pool_connection_get_timeout = *10*	(IntOpt) Number of seconds that an operation will wait to get a memcache client connection.
memcache_pool_maxsize = *10*	(IntOpt) Max total number of open connections to every memcached server. (keystone.cache.memcache_pool backend only).
memcache_pool_unused_timeout = *60*	(IntOpt) Number of seconds a connection to memcached is held unused in the pool before it is closed. (keystone.cache.memcache_pool backend only).
memcache_servers = *localhost:11211*	(ListOpt) Memcache servers in the format of "host:port". (dogpile.cache.memcache and keystone.cache.memcache_pool backends only).
memcache_socket_timeout = *3*	(IntOpt) Timeout in seconds for every call to a server. (dogpile.cache.memcache and keystone.cache.memcache_pool backends only).
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.
[memcache]	
dead_retry = *300*	(IntOpt) Number of seconds memcached server is considered dead before it is tried again. This is used by the key value store system (e.g. token pooled memcached persistence backend).
pool_connection_get_timeout = *10*	(IntOpt) Number of seconds that an operation will wait to get a memcache client connection. This is used by the key value store system (e.g. token pooled memcached persistence backend).
pool_maxsize = *10*	(IntOpt) Max total number of open connections to every memcached server. This is used by the key value store system (e.g. token pooled memcached persistence backend).
pool_unused_timeout = *60*	(IntOpt) Number of seconds a connection to memcached is held unused in the pool before it is closed. This is used by the key value store system (e.g. token pooled memcached persistence backend).

Current functional backends are:

- `dogpile.cache.memcached` - Memcached backend using the standard python-memcached library
- `dogpile.cache.pylibmc` - Memcached backend using the pylibmc library
- `dogpile.cache.bmemcached` - Memcached using python-binary-memcached library.
- `dogpile.cache.redis` - Redis backend
- `dogpile.cache.dbm` - Local DBM file backend
- `dogpile.cache.memory` - In-memory cache, not suitable for use outside of testing as it does not cleanup it's internal cache on cache expiration and does not share cache between processes. This means that caching and cache invalidation will not be consistent or reliable.
- `dogpile.cache.mongo` - MongoDB as caching backend.

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 7.2. Description of API configuration options

Configuration option = Default value	Description
[DEFAULT]	
`admin_endpoint` = *None*	(StrOpt) The base admin endpoint URL for Keystone that is advertised to clients (NOTE: this does NOT affect how Keystone listens for connections). Defaults to the base host URL of the request. E.g. a request to http://server:35357/v3/users will default to http://server:35357. You should only need to set this value if the base URL contains a path (e.g. /prefix/v3) or the endpoint should be found on a different server.
`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 AdminTokenAuthMiddleware from your paste application pipelines (for example, in keystone-paste.ini).
`compute_port` = *8774*	(IntOpt) (Deprecated) The port which the OpenStack Compute service listens on. This option was only used for string replacement in the templated catalog backend. Templated catalogs should replace the "$(compute_port)s" substitution with the static port of the compute service. As of Juno, this option is deprecated and will be removed in the L release.
`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).
`list_limit` = *None*	(IntOpt) The maximum number of entities that will be returned in a collection, 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]).
`max_param_size` = *64*	(IntOpt) Limit the sizes of user & project ID/names.
`max_project_tree_depth` = *5*	(IntOpt) Maximum depth of the project hierarchy. WARNING: setting it to a large value may adversely impact performance.
`max_token_size` = *8192*	(IntOpt) Similar to max_param_size, but provides an exception for token values.
`member_role_id` = *9fe2ff9ee4384b1894a90878d3e92bab*	(StrOpt) Similar to the member_role_name option, this represents the default role ID used to associate users with their default projects in the v2 API. This will be used as the explicit role where one is not specified by the v2 API.
`member_role_name` = *_member_*	(StrOpt) This is the role name used in combination with the member_role_id option; see that option for more detail.
`public_endpoint` = *None*	(StrOpt) The base public endpoint URL for Keystone that is advertised to clients (NOTE: this does NOT affect how Keystone listens for connections). Defaults to the base host URL of the request. E.g. a request to http://serv-

Configuration option = Default value	Description
	er:5000/v3/users will default to http://server:5000. You should only need to set this value if the base URL contains a path (e.g. /prefix/v3) or the endpoint should be found on a different server.
secure_proxy_ssl_header = *None*	(StrOpt) The HTTP header used to determine the scheme for the original request, even if it was removed by an SSL terminating proxy. Typical value is "HTTP_X_FORWARDED_PROTO".
strict_password_check = *False*	(BoolOpt) If set to true, strict password length checking is performed for password manipulation. If a password exceeds the maximum length, the operation will fail with an HTTP 403 Forbidden error. If set to false, passwords are automatically truncated to the maximum length.
[endpoint_filter]	
driver = *keystone.contrib.endpoint_filter.backends.sql.EndpointFilter*	(StrOpt) Endpoint Filter backend driver
return_all_endpoints_if_no_filter = *True*	(BoolOpt) Toggle to return all active endpoints if no filter exists.
[endpoint_policy]	
driver = *keystone.contrib.endpoint_policy.backends.sql.EndpointPolicy*	(StrOpt) Endpoint policy backend driver
[eventlet_server]	
admin_bind_host = *0.0.0.0*	(StrOpt) The IP address of the network interface for the admin service to listen on.
admin_port = *35357*	(IntOpt) The port number which the admin service listens on.
admin_workers = *None*	(IntOpt) The number of worker processes to serve the admin eventlet application. Defaults to number of CPUs (minimum of 2).
public_bind_host = *0.0.0.0*	(StrOpt) The IP address of the network interface for the public service to listen on.
public_port = *5000*	(IntOpt) The port number which the public service listens on.
public_workers = *None*	(IntOpt) The number of worker processes to serve the public eventlet application. Defaults to number of CPUs (minimum of 2).
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.
[paste_deploy]	
config_file = *keystone-paste.ini*	(StrOpt) Name of the paste configuration file that defines the available pipelines.
[resource]	
cache_time = *None*	(IntOpt) TTL (in seconds) to cache resource data. This has no effect unless global caching is enabled.
caching = *True*	(BoolOpt) Toggle for resource caching. This has no effect unless global caching is enabled.
driver = *None*	(StrOpt) Resource backend driver. If a resource driver is not specified, the assignment driver will choose the resource driver.
list_limit = *None*	(IntOpt) Maximum number of entities that will be returned in a resource collection.

Table 7.3. Description of assignment configuration options

Configuration option = Default value	Description
[assignment]	
driver = *None*	(StrOpt) Assignment backend driver.

Table 7.4. Description of authorization configuration options

Configuration option = Default value	Description
[auth]	
external = *keystone.auth.plugins.external.DefaultDomain*	(StrOpt) The external (REMOTE_USER) auth plugin module.
methods = *external, password, token, oauth1*	(ListOpt) Default auth methods.
oauth1 = *keystone.auth.plugins.oauth1.OAuth*	(StrOpt) The oAuth1.0 auth plugin module.
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 7.5. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.

Configuration option = Default value	Description
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.

Configuration option = Default value	Description
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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 7.6. Description of CA and SSL configuration options

Configuration option = Default value	Description
[eventlet_server_ssl]	
ca_certs = */etc/keystone/ssl/certs/ca.pem*	(StrOpt) Path of the CA cert file for SSL.
cert_required = *False*	(BoolOpt) Require client certificate.
certfile = */etc/keystone/ssl/certs/keystone.pem*	(StrOpt) Path of the certfile for SSL. For non-production environments, you may be interested in using `keystone-manage ssl_setup` to generate self-signed certificates.
enable = *False*	(BoolOpt) Toggle for SSL support on the Keystone eventlet servers.
keyfile = */etc/keystone/ssl/private/keystonekey.pem*	(StrOpt) Path of the keyfile for SSL.
[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. For non-production environments, you may be interested in using `keystone-manage pki_setup` to generate self-signed certificates.
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.
valid_days = *3650*	(IntOpt) Days the token signing cert is valid for (auto generated certificate).
[ssl]	
ca_key = */etc/keystone/ssl/private/cakey.pem*	(StrOpt) Path of the CA key file for SSL.
cert_subject = */C=US/ST=Unset/L=Unset/O=Unset/CN=localhost*	(StrOpt) SSL certificate subject (auto generated certificate).
key_size = *1024*	(IntOpt) SSL key length (in bits) (auto generated certificate).
valid_days = *3650*	(IntOpt) Days the certificate is valid for once signed (auto generated certificate).

Table 7.7. Description of catalog configuration options

Configuration option = Default value	Description
[catalog]	
`cache_time` = *None*	(IntOpt) Time to cache catalog data (in seconds). This has no effect unless global and catalog caching are enabled.
`caching` = *True*	(BoolOpt) Toggle for catalog caching. This has no effect unless global caching is enabled.
`driver` = *keystone.catalog.backends.sql.Catalog*	(StrOpt) 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 7.8. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
`memcached_servers` = *None*	(ListOpt) Memcached servers or None for in process cache.
[keystone_authtoken]	
`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.

Table 7.9. Description of credential configuration options

Configuration option = Default value	Description
[credential]	
`driver` = *keystone.credential.backends.sql.Credential*	(StrOpt) Credential backend driver.

Table 7.10. Description of database configuration options

Configuration option = Default value	Description
[database]	
`backend` = *sqlalchemy*	(StrOpt) The back end to use for the database.
`connection` = *None*	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between database connection retries up to db_max_retry_interval.
`db_max_retries` = *20*	(IntOpt) Maximum database connection retries before error is raised. Set to -1 to specify an infinite retry count.
`db_max_retry_interval` = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between database connection retries.
`db_retry_interval` = *1*	(IntOpt) Seconds between database 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.

Configuration option = Default value	Description
max_pool_size = *None*	(IntOpt) Maximum number of SQL connections to keep open in a pool.
max_retries = *10*	(IntOpt) Maximum number of database connection retries during startup. Set to -1 to specify 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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.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 7.11. Description of logging configuration options

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.
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 7.12. Description of domain configuration options

Configuration option = Default value	Description
[domain_config]	
cache_time = *300*	(IntOpt) TTL (in seconds) to cache domain config data. This has no effect unless domain config caching is enabled.
caching = *True*	(BoolOpt) Toggle for domain config caching. This has no effect unless global caching is enabled.
driver = *keystone.resource.config_backends.sql.DomainConfig*	(StrOpt) Domain config backend driver.

Table 7.13. Description of EC2 configuration options

Configuration option = Default value	Description
[keystone_ec2_token]	
`cafile` = *None*	(StrOpt) A PEM encoded certificate authority to use when verifying HTTPS connections. Defaults to the system CAs.
`certfile` = *None*	(StrOpt) Client certificate key filename. Required if EC2 server requires client certificate.
`insecure` = *False*	(BoolOpt) Disable SSL certificate verification.
`keyfile` = *None*	(StrOpt) Required if EC2 server requires client certificate.
`url` = *http://localhost:5000/v2.0/ec2tokens*	(StrOpt) URL to get token from ec2 request.

Table 7.14. Description of federation configuration options

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) Federation backend driver.
`federated_domain_name` = *Federated*	(StrOpt) A domain name that is reserved to allow federated ephemeral users to have a domain concept. Note that an admin will not be able to create a domain with this name or update an existing domain to this name. You are not advised to change this value unless you really have to. Changing this option to empty string or None will not have any impact and default name will be used.
`remote_id_attribute` = *None*	(StrOpt) Value to be used to obtain the entity ID of the Identity Provider from the environment (e.g. if using the mod_shib plugin this value is `Shib-Identity-Provider`).
`sso_callback_template` = */etc/keystone/sso_callback_template.html*	(StrOpt) Location of Single Sign-On callback handler, will return a token to a trusted dashboard host.
`trusted_dashboard` = *[]*	(MultiStrOpt) A list of trusted dashboard hosts. Before accepting a Single Sign-On request to return a token, the origin host must be a member of the trusted_dashboard list. This configuration option may be repeated for multiple values. For example: trusted_dashboard=http://acme.com trusted_dashboard=http://beta.com

Table 7.15. Description of Fernet tokens configuration options

Configuration option = Default value	Description
[fernet_tokens]	
`key_repository` = */etc/keystone/fernet-keys/*	(StrOpt) Directory containing Fernet token keys.
`max_active_keys` = *3*	(IntOpt) This controls how many keys are held in rotation by keystone-manage fernet_rotate before they are discarded. The default value of 3 means that keystone will maintain one staged key, one primary key, and one secondary key. Increasing this value means that additional secondary keys will be kept in the rotation.

Table 7.16. Description of identity configuration options

Configuration option = Default value	Description
[identity]	
`cache_time` = *600*	(IntOpt) Time to cache identity data (in seconds). This has no effect unless global and identity caching are enabled.

Configuration option = Default value	Description
caching = *True*	(BoolOpt) Toggle for identity caching. This has no effect unless global caching is enabled.
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.
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_configurations_from_database = *False*	(BoolOpt) Extract the domain specific configuration options from the resource backend where they have been stored with the domain data. This feature is disabled by default (in which case the domain specific options will be loaded from files in the domain configuration directory); set to true to enable.
domain_specific_drivers_enabled = *False*	(BoolOpt) A subset (or all) of domains can have their own identity driver, each with their own partial configuration options, stored in either the resource backend or in a file in a domain configuration directory (depending on the setting of domain_configurations_from_database). Only values specific to the domain need to be specified in this manner. This feature is disabled by default; set to true to enable.
driver = *keystone.identity.backends.sql.Identity*	(StrOpt) 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 7.17. Description of KVS configuration options

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 (in seconds) 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 7.18. Description of LDAP configuration options

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

Configuration option = Default value	Description
	"default". The "default" option falls back to using default dereferencing configured by your ldap.conf.
allow_subtree_delete = *False*	(BoolOpt) Delete subtrees using the subtree delete control. Only enable this option if your LDAP server supports subtree deletion.
auth_pool_connection_lifetime = *60*	(IntOpt) End user auth connection lifetime in seconds.
auth_pool_size = *100*	(IntOpt) End user auth connection pool size.
chase_referrals = *None*	(BoolOpt) Override the system's default referral chasing behavior for queries.
debug_level = *None*	(IntOpt) Sets the LDAP debugging level for LDAP calls. A value of 0 means that debugging is not enabled. This value is a bitmask, consult your LDAP documentation for possible values.
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.
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.
pool_connection_lifetime = *600*	(IntOpt) Connection lifetime in seconds.
pool_connection_timeout = *-1*	(IntOpt) Connector timeout in seconds. Value -1 indicates indefinite wait for response.
pool_retry_delay = *0.1*	(FloatOpt) Time span in seconds to wait between two reconnect trials.
pool_retry_max = *3*	(IntOpt) Maximum count of reconnect trials.
pool_size = *10*	(IntOpt) Connection pool size.
project_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.
project_allow_create = *True*	(BoolOpt) Allow project creation in LDAP backend.
project_allow_delete = *True*	(BoolOpt) Allow project deletion in LDAP backend.
project_allow_update = *True*	(BoolOpt) Allow project update in LDAP backend.

Configuration option = Default value	Description
project_attribute_ignore =	(ListOpt) List of attributes stripped off the project on update.
project_desc_attribute = *description*	(StrOpt) LDAP attribute mapped to project description.
project_domain_id_attribute = *businessCategory*	(StrOpt) LDAP attribute mapped to project domain_id.
project_enabled_attribute = *enabled*	(StrOpt) LDAP attribute mapped to project enabled.
project_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 "project_enabled_emulation_dn" group.
project_enabled_emulation_dn = *None*	(StrOpt) DN of the group entry to hold enabled projects when using enabled emulation.
project_filter = *None*	(StrOpt) LDAP search filter for projects.
project_id_attribute = *cn*	(StrOpt) LDAP attribute mapped to project id.
project_member_attribute = *member*	(StrOpt) LDAP attribute mapped to project membership for user.
project_name_attribute = *ou*	(StrOpt) LDAP attribute mapped to project name.
project_objectclass = *groupOfNames*	(StrOpt) LDAP objectclass for projects.
project_tree_dn = *None*	(StrOpt) Search base for projects
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
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_auth_pool = *False*	(BoolOpt) Enable LDAP connection pooling for end user authentication. If use_pool is disabled, then this setting is meaningless and is not used at all.
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_pool = *False*	(BoolOpt) Enable LDAP connection pooling.

Configuration option = Default value	Description
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_invert = *False*	(BoolOpt) Invert the meaning of the boolean enabled values. Some LDAP servers use a boolean lock attribute where "true" means an account is disabled. Setting "user_enabled_invert = true" will allow these lock attributes to be used. This setting will have no effect if "user_enabled_mask" or "user_enabled_emulation" settings are in use.
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. WARNING: must not be a multivalued attribute.
user_mail_attribute = *mail*	(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.
user_pass_attribute = *userPassword*	(StrOpt) LDAP attribute mapped to password.
user_tree_dn = *None*	(StrOpt) Search base for users.

Table 7.19. Description of logging configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
use_stderr = *True*	(BoolOpt) Log output to standard error.

Configuration option = Default value	Description
verbose = *False*	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 7.20. Description of mapping configuration options

Configuration option = Default value	Description
[identity_mapping]	
backward_compatible_ids = *True*	(BoolOpt) The format of user and group IDs changed in Juno for backends that do not generate UUIDs (e.g. LDAP), with keystone providing a hash mapping to the underlying attribute in LDAP. By default this mapping is disabled, which ensures that existing IDs will not change. Even when the mapping is enabled by using domain specific drivers, any users and groups from the default domain being handled by LDAP will still not be mapped to ensure their IDs remain backward compatible. Setting this value to False will enable the mapping for even the default LDAP driver. It is only safe to do this if you do not already have assignments for users and groups from the default LDAP domain, and it is acceptable for Keystone to provide the different IDs to clients than it did previously. Typically this means that the only time you can set this value to False is when configuring a fresh installation.
driver = *keystone.identity.mapping_backends.sql.Mapping*	(StrOpt) Keystone Identity Mapping backend driver.
generator = *keystone.identity.id_generators.sha256.Generator*	(StrOpt) Public ID generator for user and group entities. The Keystone identity mapper only supports generators that produce no more than 64 characters.

Table 7.21. Description of memcache configuration options

Configuration option = Default value	Description
[memcache]	
servers = *localhost:11211*	(ListOpt) Memcache servers in the format of "host:port".
socket_timeout = *3*	(IntOpt) Timeout in seconds for every call to a server. This is used by the key value store system (e.g. token pooled memcached persistence backend).

Table 7.22. Description of OAuth configuration options

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) Credential backend driver.
request_token_duration = *28800*	(IntOpt) Duration (in seconds) for the OAuth Request Token.

Table 7.23. Description of os_inherit configuration options

Configuration option = Default value	Description
[os_inherit]	
enabled = *False*	(BoolOpt) role-assignment inheritance to projects from owning domain or from projects higher in the hierarchy can be optionally enabled.

Table 7.24. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
`max_request_body_size = 114688`	(IntOpt) The maximum body size for each request, in bytes.

Table 7.25. Description of policy configuration options

Configuration option = Default value	Description
[oslo_policy]	
`policy_default_rule = default`	(StrOpt) Default rule. Enforced when a requested rule is not found.
`policy_dirs = ['policy.d']`	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
`policy_file = policy.json`	(StrOpt) The JSON file that defines policies.
[policy]	
`driver = keystone.policy.backends.sql.Policy`	(StrOpt) Policy backend driver.
`list_limit = None`	(IntOpt) Maximum number of entities that will be returned in a policy collection.

Table 7.26. Description of revoke configuration options

Configuration option = Default value	Description
[revoke]	
`cache_time = 3600`	(IntOpt) Time to cache the revocation list and the revocation events (in seconds). This has no effect unless global and token caching are enabled.
`caching = True`	(BoolOpt) Toggle for revocation event caching. This has no effect unless global caching is enabled.
`driver = keystone.contrib.revoke.backends.sql.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 7.27. Description of role configuration options

Configuration option = Default value	Description
[role]	
`cache_time = None`	(IntOpt) TTL (in seconds) to cache role data. This has no effect unless global caching is enabled.
`caching = True`	(BoolOpt) Toggle for role caching. This has no effect unless global caching is enabled.
`driver = None`	(StrOpt) Role backend driver.
`list_limit = None`	(IntOpt) Maximum number of entities that will be returned in a role collection.

Table 7.28. Description of SAML configuration options

Configuration option = Default value	Description
[saml]	
assertion_expiration_time = *3600*	(IntOpt) Default TTL, in seconds, for any generated SAML assertion created by Keystone.
certfile = */etc/keystone/ssl/certs/signing_cert.pem*	(StrOpt) Path of the certfile for SAML signing. For non-production environments, you may be interested in using `keystone-manage pki_setup` to generate self-signed certificates. Note, the path cannot contain a comma.
idp_contact_company = *None*	(StrOpt) Company of contact person.
idp_contact_email = *None*	(StrOpt) Email address of contact person.
idp_contact_name = *None*	(StrOpt) Given name of contact person
idp_contact_surname = *None*	(StrOpt) Surname of contact person.
idp_contact_telephone = *None*	(StrOpt) Telephone number of contact person.
idp_contact_type = *other*	(StrOpt) Contact type. Allowed values are: technical, support, administrative billing, and other
idp_entity_id = *None*	(StrOpt) Entity ID value for unique Identity Provider identification. Usually FQDN is set with a suffix. A value is required to generate IDP Metadata. For example: https://keystone.example.com/v3/OS-FEDERATION/saml2/idp
idp_lang = *en*	(StrOpt) Language used by the organization.
idp_metadata_path = */etc/keystone/saml2_idp_metadata.xml*	(StrOpt) Path to the Identity Provider Metadata file. This file should be generated with the keystone-manage saml_idp_metadata command.
idp_organization_display_name = *None*	(StrOpt) Organization name to be displayed.
idp_organization_name = *None*	(StrOpt) Organization name the installation belongs to.
idp_organization_url = *None*	(StrOpt) URL of the organization.
idp_sso_endpoint = *None*	(StrOpt) Identity Provider Single-Sign-On service value, required in the Identity Provider's metadata. A value is required to generate IDP Metadata. For example: https://keystone.example.com/v3/OS-FEDERATION/saml2/sso
keyfile = */etc/keystone/ssl/private/signing_key.pem*	(StrOpt) Path of the keyfile for SAML signing. Note, the path cannot contain a comma.
relay_state_prefix = *ss:mem:*	(StrOpt) The prefix to use for the RelayState SAML attribute, used when generating ECP wrapped assertions.
xmlsec1_binary = *xmlsec1*	(StrOpt) Binary to be called for XML signing. Install the appropriate package, specify absolute path or adjust your PATH environment variable if the binary cannot be found.

Table 7.29. Description of security configuration options

Configuration option = Default value	Description
[DEFAULT]	
crypt_strength = *40000*	(IntOpt) The value passed as the keyword "rounds" to passlib's encrypt method.

Table 7.30. Description of token configuration options

Configuration option = Default value	Description
[token]	
allow_rescope_scoped_token = *True*	(BoolOpt) Allow rescoping of scoped token. Setting allow_rescoped_scoped_token to false prevents a user from exchanging a scoped token for any other token.

Configuration option = Default value	Description
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.persistence.backends.sql.Token*	(StrOpt) 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).
hash_algorithm = *md5*	(StrOpt) The hash algorithm to use for PKI tokens. This can be set to any algorithm that hashlib supports. WARNING: Before changing this value, the auth_token middleware must be configured with the hash_algorithms, otherwise token revocation will not be processed correctly.
provider = *keystone.token.providers.uuid.Provider*	(StrOpt) Controls the token construction, validation, and revocation operations. Core providers are "keystone.token.providers.[fernet|pkiz|pki|uuid].Provider".
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 7.31. Description of trust configuration options

Configuration option = Default value	Description
[trust]	
allow_redelegation = *False*	(BoolOpt) Enable redelegation feature.
driver = *keystone.trust.backends.sql.Trust*	(StrOpt) Trust backend driver.
enabled = *True*	(BoolOpt) Delegation and impersonation features can be optionally disabled.
max_redelegation_count = *3*	(IntOpt) Maximum depth of trust redelegation.

Table 7.32. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.

Configuration option = Default value	Description
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 7.33. Description of AMQP configuration options

Configuration option = Default value	Description
[DEFAULT]	
control_exchange = *keystone*	(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_format = *basic*	(StrOpt) Define the notification format for Identity Service events. A "basic" notification has information about the resource being operated on. A "cadf" notification has the same information, as well as information about the initiator of the event. Valid options are: basic and cadf
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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 7.34. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
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.

Configuration option = Default value	Description
qpid_protocol = *tcp*	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
qpid_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 7.35. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
rabbit_retry_backoff = *2*	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.

Configuration option = Default value	Description
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_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

Table 7.36. Description of ZeroMQ configuration options

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 = localhost	(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 = local	(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 7.37. Description of Redis configuration options

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).

Identity service sample configuration files

You can find the files described in this section in the `/etc/keystone` directory.

keystone.conf

Use the `keystone.conf` file to configure most Identity service options:

```
[DEFAULT]

#
# From 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

# (Deprecated) The port which the OpenStack Compute service listens on. This
# option was only used for string replacement in the templated catalog
 backend.
# Templated catalogs should replace the "$(compute_port)s" substitution with
# the static port of the compute service. As of Juno, this option is
 deprecated
# and will be removed in the L release. (integer value)
#compute_port = 8774

# The base public endpoint URL for Keystone that is advertised to clients
# (NOTE: this does NOT affect how Keystone listens for connections). Defaults
# to the base host URL of the request. E.g. a request to
# http://server:5000/v3/users will default to http://server:5000. You should
# only need to set this value if the base URL contains a path (e.g. /prefix/
v3)
# or the endpoint should be found on a different server. (string value)
#public_endpoint = <None>

# The base admin endpoint URL for Keystone that is advertised to clients
 (NOTE:
# this does NOT affect how Keystone listens for connections). Defaults to the
# base host URL of the request. E.g. a request to http://server:35357/v3/users
# will default to http://server:35357. You should only need to set this value
# if the base URL contains a path (e.g. /prefix/v3) or the endpoint should be
# found on a different server. (string value)
#admin_endpoint = <None>

# Maximum depth of the project hierarchy. WARNING: setting it to a large value
# may adversely impact performance. (integer value)
#max_project_tree_depth = 5

# Limit the sizes of user & project ID/names. (integer value)
#max_param_size = 64

# Similar to max_param_size, but provides an exception for token values.
# (integer value)
#max_token_size = 8192

# Similar to the member_role_name option, this represents the default role ID
# used to associate users with their default projects in the v2 API. This will
# be used as the explicit role where one is not specified by the v2 API.
# (string value)
#member_role_id = 9fe2ff9ee4384b1894a90878d3e92bab

# This is the role name used in combination with the member_role_id option;
 see
# that option for more detail. (string value)
#member_role_name = _member_

# The value passed as the keyword "rounds" to passlib's encrypt method.
# (integer value)
#crypt_strength = 40000

# The maximum number of entities that will be returned in a collection, 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

# If set to true, strict password length checking is performed for password
# manipulation. If a password exceeds the maximum length, the operation will
# fail with an HTTP 403 Forbidden error. If set to false, passwords are
# automatically truncated to the maximum length. (boolean value)
#strict_password_check = false

# The HTTP header used to determine the scheme for the original request, even
# if it was removed by an SSL terminating proxy. Typical value is
# "HTTP_X_FORWARDED_PROTO". (string value)
#secure_proxy_ssl_header = <None>

#
# From keystone.notifications
#

# Default publisher_id for outgoing notifications (string value)
#default_publisher_id = <None>

# Define the notification format for Identity Service events. A "basic"
# notification has information about the resource being operated on. A "cadf"
# notification has the same information, as well as information about the
# initiator of the event. Valid options are: basic and cadf (string value)
#notification_format = basic

#
# From keystone.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>

#
# From oslo.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
```

```
# The name of a logging configuration file. This file is appended to any
# existing logging configuration files. For details about logging
 configuration
# files, see the Python logging module documentation. (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
# will change in J to honor RFC5424. (boolean value)
#use_syslog = false

# (Optional) Enables or disables syslog rfc5424 format for logging. If
 enabled,
# prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The
# format without the 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

# 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,amqplib=WARN,boto=WARN,qpid=WARN,sqlalchemy=
WARN,suds=INFO,oslo.messaging=INFO,iso8601=WARN,requests.packages.urllib3.
connectionpool=WARN,urllib3.connectionpool=WARN,websocket=WARN,requests.
packages.urllib3.util.retry=WARN,urllib3.util.retry=WARN,keystonemiddleware=
WARN,routes.middleware=WARN,stevedore=WARN

# Enables or disables publication of error events. (boolean value)
#publish_errors = false

# Enables or disables fatal status of deprecations. (boolean value)
#fatal_deprecations = false

# The format for an instance that is passed with the log message. (string
# value)
#instance_format = "[instance: %(uuid)s] "

# The format for an instance UUID that is passed with the log message. (string
# value)
#instance_uuid_format = "[instance: %(uuid)s] "

#
# From oslo.messaging
#

# 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 = localhost

# 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 thread 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 = keystone


[assignment]

#
# From keystone
#

# Assignment backend driver. (string value)
#driver = <None>


[auth]

#
# From keystone
#

# Default auth methods. (list value)
#methods = external,password,token,oauth1

# 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

# The oAuth1.0 auth plugin module. (string value)
#oauth1 = keystone.auth.plugins.oauth1.OAuth
```

```
[cache]

#
# From 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 with pooling
# (keystone.cache.memcache_pool) 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

# 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.
# (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

# Memcache servers in the format of "host:port". (dogpile.cache.memcache and
# keystone.cache.memcache_pool backends only). (list value)
#memcache_servers = localhost:11211

# Number of seconds memcached server is considered dead before it is tried
# again. (dogpile.cache.memcache and keystone.cache.memcache_pool backends
# only). (integer value)
#memcache_dead_retry = 300

# Timeout in seconds for every call to a server. (dogpile.cache.memcache and
# keystone.cache.memcache_pool backends only). (integer value)
#memcache_socket_timeout = 3

# Max total number of open connections to every memcached server.
# (keystone.cache.memcache_pool backend only). (integer value)
#memcache_pool_maxsize = 10
```

```
# Number of seconds a connection to memcached is held unused in the pool
 before
# it is closed. (keystone.cache.memcache_pool backend only). (integer value)
#memcache_pool_unused_timeout = 60

# Number of seconds that an operation will wait to get a memcache client
# connection. (integer value)
#memcache_pool_connection_get_timeout = 10


[catalog]

#
# From keystone
#

# Catalog template file name for use with the template catalog backend.
 (string
# value)
#template_file = default_catalog.templates

# Catalog backend driver. (string value)
#driver = keystone.catalog.backends.sql.Catalog

# Toggle for catalog caching. This has no effect unless global caching is
# enabled. (boolean value)
#caching = true

# Time to cache catalog data (in seconds). This has no effect unless global
 and
# catalog caching are enabled. (integer value)
#cache_time = <None>

# Maximum number of entities that will be returned in a catalog collection.
# (integer value)
#list_limit = <None>


[credential]

#
# From keystone
#

# Credential backend driver. (string value)
#driver = keystone.credential.backends.sql.Credential


[database]

#
# From oslo.db
#

# The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
#sqlite_db = oslo.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
```

```
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use 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 SQLAlchemy connection string to use to connect to the slave database.
# (string value)
#slave_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 number of database connection retries during startup. Set to -1 to
# specify 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 retries of a database transaction. (integer value)
#db_retry_interval = 1

# If True, increases the interval between retries of a database operation up
 to
# db_max_retry_interval. (boolean value)
#db_inc_retry_interval = true

# If db_inc_retry_interval is set, the maximum seconds between retries of a
# database operation. (integer value)
#db_max_retry_interval = 10

# Maximum retries in case of connection error or deadlock error before error
 is
# raised. Set to -1 to specify an infinite retry count. (integer value)
#db_max_retries = 20


[domain_config]

#
# From keystone
#

# Domain config backend driver. (string value)
#driver = keystone.resource.config_backends.sql.DomainConfig

# Toggle for domain config caching. This has no effect unless global caching
 is
# enabled. (boolean value)
#caching = true

# TTL (in seconds) to cache domain config data. This has no effect unless
# domain config caching is enabled. (integer value)
#cache_time = 300


[endpoint_filter]

#
# From 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
```

```
[endpoint_policy]

#
# From keystone
#

# Endpoint policy backend driver (string value)
#driver = keystone.contrib.endpoint_policy.backends.sql.EndpointPolicy


[eventlet_server]

#
# From keystone
#

# The number of worker processes to serve the public eventlet application.
# Defaults to number of CPUs (minimum of 2). (integer value)
# Deprecated group/name - [DEFAULT]/public_workers
#public_workers = <None>

# The number of worker processes to serve the admin eventlet application.
# Defaults to number of CPUs (minimum of 2). (integer value)
# Deprecated group/name - [DEFAULT]/admin_workers
#admin_workers = <None>

# The IP address of the network interface for the public service to listen on.
# (string value)
# Deprecated group/name - [DEFAULT]/bind_host
# Deprecated group/name - [DEFAULT]/public_bind_host
#public_bind_host = 0.0.0.0

# The port number which the public service listens on. (integer value)
# Deprecated group/name - [DEFAULT]/public_port
#public_port = 5000

# The IP address of the network interface for the admin service to listen on.
# (string value)
# Deprecated group/name - [DEFAULT]/bind_host
# Deprecated group/name - [DEFAULT]/admin_bind_host
#admin_bind_host = 0.0.0.0

# The port number which the admin service listens on. (integer value)
# Deprecated group/name - [DEFAULT]/admin_port
#admin_port = 35357

# 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. (boolean
# value)
# Deprecated group/name - [DEFAULT]/tcp_keepalive
#tcp_keepalive = false

# Sets the value of TCP_KEEPIDLE in seconds for each server socket. Only
# applies if tcp_keepalive is true. (integer value)
# Deprecated group/name - [DEFAULT]/tcp_keepidle
#tcp_keepidle = 600
```

```
[eventlet_server_ssl]

#
# From keystone
#

# Toggle for SSL support on the Keystone eventlet servers. (boolean value)
# Deprecated group/name - [ssl]/enable
#enable = false

# Path of the certfile for SSL. For non-production environments, you may be
# interested in using `keystone-manage ssl_setup` to generate self-signed
# certificates. (string value)
# Deprecated group/name - [ssl]/certfile
#certfile = /etc/keystone/ssl/certs/keystone.pem

# Path of the keyfile for SSL. (string value)
# Deprecated group/name - [ssl]/keyfile
#keyfile = /etc/keystone/ssl/private/keystonekey.pem

# Path of the CA cert file for SSL. (string value)
# Deprecated group/name - [ssl]/ca_certs
#ca_certs = /etc/keystone/ssl/certs/ca.pem

# Require client certificate. (boolean value)
# Deprecated group/name - [ssl]/cert_required
#cert_required = false


[federation]

#
# From 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 =

# Value to be used to obtain the entity ID of the Identity Provider from the
# environment (e.g. if using the mod_shib plugin this value is `Shib-Identity-
# Provider`). (string value)
#remote_id_attribute = <None>

# A domain name that is reserved to allow federated ephemeral users to have a
# domain concept. Note that an admin will not be able to create a domain with
# this name or update an existing domain to this name. You are not advised to
# change this value unless you really have to. Changing this option to empty
# string or None will not have any impact and default name will be used.
# (string value)
#federated_domain_name = Federated

# A list of trusted dashboard hosts. Before accepting a Single Sign-On request
# to return a token, the origin host must be a member of the trusted_dashboard
# list. This configuration option may be repeated for multiple values. For
# example: trusted_dashboard=http://acme.com trusted_dashboard=http://beta.com
# (multi valued)
```

```
#trusted_dashboard =

# Location of Single Sign-On callback handler, will return a token to a
 trusted
# dashboard host. (string value)
#sso_callback_template = /etc/keystone/sso_callback_template.html


[fernet_tokens]

#
# From keystone
#

# Directory containing Fernet token keys. (string value)
#key_repository = /etc/keystone/fernet-keys/

# This controls how many keys are held in rotation by keystone-manage
# fernet_rotate before they are discarded. The default value of 3 means that
# keystone will maintain one staged key, one primary key, and one secondary
# key. Increasing this value means that additional secondary keys will be kept
# in the rotation. (integer value)
#max_active_keys = 3


[identity]

#
# From 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 options, stored in either the resource
# backend or in a file in a domain configuration directory (depending on the
# setting of domain_configurations_from_database). Only values specific to the
# domain need to be specified in this manner. This feature is disabled by
# default; set to true to enable. (boolean value)
#domain_specific_drivers_enabled = false

# Extract the domain specific configuration options from the resource backend
# where they have been stored with the domain data. This feature is disabled
 by
# default (in which case the domain specific options will be loaded from files
# in the domain configuration directory); set to true to enable. (boolean
# value)
#domain_configurations_from_database = 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
```

```
# Identity backend driver. (string value)
#driver = keystone.identity.backends.sql.Identity

# Toggle for identity caching. This has no effect unless global caching is
# enabled. (boolean value)
#caching = true

# Time to cache identity data (in seconds). This has no effect unless global
# and identity caching are enabled. (integer value)
#cache_time = 600

# 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>


[identity_mapping]

#
# From keystone
#

# Keystone Identity Mapping backend driver. (string value)
#driver = keystone.identity.mapping_backends.sql.Mapping

# Public ID generator for user and group entities. The Keystone identity
 mapper
# only supports generators that produce no more than 64 characters. (string
# value)
#generator = keystone.identity.id_generators.sha256.Generator

# The format of user and group IDs changed in Juno for backends that do not
# generate UUIDs (e.g. LDAP), with keystone providing a hash mapping to the
# underlying attribute in LDAP. By default this mapping is disabled, which
# ensures that existing IDs will not change. Even when the mapping is enabled
# by using domain specific drivers, any users and groups from the default
# domain being handled by LDAP will still not be mapped to ensure their IDs
# remain backward compatible. Setting this value to False will enable the
# mapping for even the default LDAP driver. It is only safe to do this if you
# do not already have assignments for users and groups from the default LDAP
# domain, and it is acceptable for Keystone to provide the different IDs to
# clients than it did previously.  Typically this means that the only time you
# can set this value to False is when configuring a fresh installation.
# (boolean value)
#backward_compatible_ids = true


[kvs]

#
# From 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 (in seconds) for distributed locking. (integer value)
#default_lock_timeout = 5


[ldap]

#
# From 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

# Delete subtrees using the subtree delete control. Only enable this option if
# your LDAP server supports subtree deletion. (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
```

```
# Sets the LDAP debugging level for LDAP calls. A value of 0 means that
# debugging is not enabled. This value is a bitmask, consult your LDAP
# documentation for possible values. (integer value)
#debug_level = <None>

# 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. WARNING: must not be a multivalued
# attribute. (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 = mail

# LDAP attribute mapped to password. (string value)
#user_pass_attribute = userPassword

# LDAP attribute mapped to user enabled flag. (string value)
#user_enabled_attribute = enabled

# Invert the meaning of the boolean enabled values. Some LDAP servers use a
# boolean lock attribute where "true" means an account is disabled. Setting
# "user_enabled_invert = true" will allow these lock attributes to be used.
# This setting will have no effect if "user_enabled_mask" or
# "user_enabled_emulation" settings are in use. (boolean value)
#user_enabled_invert = false

# 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)
# Deprecated group/name - [ldap]/tenant_tree_dn
#project_tree_dn = <None>

# LDAP search filter for projects. (string value)
# Deprecated group/name - [ldap]/tenant_filter
#project_filter = <None>

# LDAP objectclass for projects. (string value)
# Deprecated group/name - [ldap]/tenant_objectclass
#project_objectclass = groupOfNames

# LDAP attribute mapped to project id. (string value)
# Deprecated group/name - [ldap]/tenant_id_attribute
#project_id_attribute = cn

# LDAP attribute mapped to project membership for user. (string value)
# Deprecated group/name - [ldap]/tenant_member_attribute
#project_member_attribute = member

# LDAP attribute mapped to project name. (string value)
# Deprecated group/name - [ldap]/tenant_name_attribute
#project_name_attribute = ou

# LDAP attribute mapped to project description. (string value)
# Deprecated group/name - [ldap]/tenant_desc_attribute
#project_desc_attribute = description

# LDAP attribute mapped to project enabled. (string value)
# Deprecated group/name - [ldap]/tenant_enabled_attribute
#project_enabled_attribute = enabled
```

```
# LDAP attribute mapped to project domain_id. (string value)
# Deprecated group/name - [ldap]/tenant_domain_id_attribute
#project_domain_id_attribute = businessCategory

# List of attributes stripped off the project on update. (list value)
# Deprecated group/name - [ldap]/tenant_attribute_ignore
#project_attribute_ignore =

# Allow project creation in LDAP backend. (boolean value)
# Deprecated group/name - [ldap]/tenant_allow_create
#project_allow_create = true

# Allow project update in LDAP backend. (boolean value)
# Deprecated group/name - [ldap]/tenant_allow_update
#project_allow_update = true

# Allow project deletion in LDAP backend. (boolean value)
# Deprecated group/name - [ldap]/tenant_allow_delete
#project_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
# "project_enabled_emulation_dn" group. (boolean value)
# Deprecated group/name - [ldap]/tenant_enabled_emulation
#project_enabled_emulation = false

# DN of the group entry to hold enabled projects when using enabled emulation.
# (string value)
# Deprecated group/name - [ldap]/tenant_enabled_emulation_dn
#project_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)
# Deprecated group/name - [ldap]/tenant_additional_attribute_mapping
#project_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

# Enable LDAP connection pooling. (boolean value)
#use_pool = false

# Connection pool size. (integer value)
#pool_size = 10

# Maximum count of reconnect trials. (integer value)
#pool_retry_max = 3

# Time span in seconds to wait between two reconnect trials. (floating point
# value)
#pool_retry_delay = 0.1

# Connector timeout in seconds. Value -1 indicates indefinite wait for
# response. (integer value)
#pool_connection_timeout = -1

# Connection lifetime in seconds. (integer value)
#pool_connection_lifetime = 600

# Enable LDAP connection pooling for end user authentication. If use_pool is
# disabled, then this setting is meaningless and is not used at all. (boolean
# value)
#use_auth_pool = false

# End user auth connection pool size. (integer value)
#auth_pool_size = 100

# End user auth connection lifetime in seconds. (integer value)
#auth_pool_connection_lifetime = 60


[matchmaker_redis]

#
# From 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]

#
# From oslo.messaging
#

# Matchmaker ring file (JSON). (string value)
# Deprecated group/name - [DEFAULT]/matchmaker_ringfile
```

```
#ringfile = /etc/oslo/matchmaker_ring.json


[memcache]

#
# From keystone
#

# Memcache servers in the format of "host:port". (list value)
#servers = localhost:11211

# Number of seconds memcached server is considered dead before it is tried
# again. This is used by the key value store system (e.g. token pooled
# memcached persistence backend). (integer value)
#dead_retry = 300

# Timeout in seconds for every call to a server. This is used by the key value
# store system (e.g. token pooled memcached persistence backend). (integer
# value)
#socket_timeout = 3

# Max total number of open connections to every memcached server. This is used
# by the key value store system (e.g. token pooled memcached persistence
# backend). (integer value)
#pool_maxsize = 10

# Number of seconds a connection to memcached is held unused in the pool
 before
# it is closed. This is used by the key value store system (e.g. token pooled
# memcached persistence backend). (integer value)
#pool_unused_timeout = 60

# Number of seconds that an operation will wait to get a memcache client
# connection. This is used by the key value store system (e.g. token pooled
# memcached persistence backend). (integer value)
#pool_connection_get_timeout = 10


[oauth1]

#
# From 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]

#
# From keystone
#
```

```
# role-assignment inheritance to projects from owning domain or from projects
# higher in the hierarchy can be optionally enabled. (boolean value)
#enabled = false


[oslo_messaging_amqp]

#
# From oslo.messaging
#

# address prefix used when sending to a specific server (string value)
# Deprecated group/name - [amqp1]/server_request_prefix
#server_request_prefix = exclusive

# address prefix used when broadcasting to all servers (string value)
# Deprecated group/name - [amqp1]/broadcast_prefix
#broadcast_prefix = broadcast

# address prefix when sending to any server in group (string value)
# Deprecated group/name - [amqp1]/group_request_prefix
#group_request_prefix = unicast

# Name for the AMQP container (string value)
# Deprecated group/name - [amqp1]/container_name
#container_name = <None>

# Timeout for inactive connections (in seconds) (integer value)
# Deprecated group/name - [amqp1]/idle_timeout
#idle_timeout = 0

# Debug: dump AMQP frames to stdout (boolean value)
# Deprecated group/name - [amqp1]/trace
#trace = false

# CA certificate PEM file for verifing server certificate (string value)
# Deprecated group/name - [amqp1]/ssl_ca_file
#ssl_ca_file =

# Identifying certificate PEM file to present to clients (string value)
# Deprecated group/name - [amqp1]/ssl_cert_file
#ssl_cert_file =

# Private key PEM file used to sign cert_file certificate (string value)
# Deprecated group/name - [amqp1]/ssl_key_file
#ssl_key_file =

# Password for decrypting ssl_key_file (if encrypted) (string value)
# Deprecated group/name - [amqp1]/ssl_key_password
#ssl_key_password = <None>

# Accept clients using either SSL or plain TCP (boolean value)
# Deprecated group/name - [amqp1]/allow_insecure_clients
#allow_insecure_clients = false


[oslo_messaging_qpid]

#
```

```
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
#amqp_auto_delete = false

# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
#rpc_conn_pool_size = 30

# Qpid broker hostname. (string value)
# Deprecated group/name - [DEFAULT]/qpid_hostname
#qpid_hostname = localhost

# Qpid broker port. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_port
#qpid_port = 5672

# Qpid HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/qpid_hosts
#qpid_hosts = $qpid_hostname:$qpid_port

# Username for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_username
#qpid_username =

# Password for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_password
#qpid_password =

# Space separated list of SASL mechanisms to use for auth. (string value)
# Deprecated group/name - [DEFAULT]/qpid_sasl_mechanisms
#qpid_sasl_mechanisms =

# Seconds between connection keepalive heartbeats. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_heartbeat
#qpid_heartbeat = 60

# Transport to use, either 'tcp' or 'ssl'. (string value)
# Deprecated group/name - [DEFAULT]/qpid_protocol
#qpid_protocol = tcp

# Whether to disable the Nagle algorithm. (boolean value)
# Deprecated group/name - [DEFAULT]/qpid_tcp_nodelay
#qpid_tcp_nodelay = true

# The number of prefetched messages held by receiver. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_receiver_capacity
#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)
```

```
# Deprecated group/name - [DEFAULT]/qpid_topology_version
#qpid_topology_version = 1


[oslo_messaging_rabbit]

#
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
#amqp_auto_delete = false

# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
#rpc_conn_pool_size = 30

# SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and
# SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some
# distributions. (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_version
#kombu_ssl_version =

# SSL key file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_keyfile
#kombu_ssl_keyfile =

# SSL cert file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_certfile
#kombu_ssl_certfile =

# SSL certification authority file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_ca_certs
#kombu_ssl_ca_certs =

# How long to wait before reconnecting in response to an AMQP consumer cancel
# notification. (floating point value)
# Deprecated group/name - [DEFAULT]/kombu_reconnect_delay
#kombu_reconnect_delay = 1.0

# The RabbitMQ broker address where a single node is used. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_host
#rabbit_host = localhost

# The RabbitMQ broker port where a single node is used. (integer value)
# Deprecated group/name - [DEFAULT]/rabbit_port
#rabbit_port = 5672

# RabbitMQ HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/rabbit_hosts
#rabbit_hosts = $rabbit_host:$rabbit_port

# Connect over SSL for RabbitMQ. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_use_ssl
#rabbit_use_ssl = false
```

```
# The RabbitMQ userid. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_userid
#rabbit_userid = guest

# The RabbitMQ password. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_password
#rabbit_password = guest

# The RabbitMQ login method. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_login_method
#rabbit_login_method = AMQPLAIN

# The RabbitMQ virtual host. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_virtual_host
#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)
# Deprecated group/name - [DEFAULT]/rabbit_retry_backoff
#rabbit_retry_backoff = 2

# Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry
# count). (integer value)
# Deprecated group/name - [DEFAULT]/rabbit_max_retries
#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)
# Deprecated group/name - [DEFAULT]/rabbit_ha_queues
#rabbit_ha_queues = false

# Number of seconds after which the Rabbit broker is considered down if
# heartbeat's keep-alive fails (0 disable the heartbeat). (integer value)
#heartbeat_timeout_threshold = 60

# How often times during the heartbeat_timeout_threshold we check the
# heartbeat. (integer value)
#heartbeat_rate = 2

# Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake (boolean value)
# Deprecated group/name - [DEFAULT]/fake_rabbit
#fake_rabbit = false


[oslo_middleware]

#
# From oslo.middleware
#

# The maximum body size for each  request, in bytes. (integer value)
# Deprecated group/name - [DEFAULT]/osapi_max_request_body_size
# Deprecated group/name - [DEFAULT]/max_request_body_size
#max_request_body_size = 114688
```

```
[oslo_policy]

#
# From oslo.policy
#

# The JSON file that defines policies. (string value)
# Deprecated group/name - [DEFAULT]/policy_file
#policy_file = policy.json

# Default rule. Enforced when a requested rule is not found. (string value)
# Deprecated group/name - [DEFAULT]/policy_default_rule
#policy_default_rule = default

# Directories where policy configuration files are stored. They can be
 relative
# to any directory in the search path defined by the config_dir option, or
# absolute paths. The file defined by policy_file must exist for these
# directories to be searched.  Missing or empty directories are ignored.
 (multi
# valued)
# Deprecated group/name - [DEFAULT]/policy_dirs
#policy_dirs = policy.d


[paste_deploy]

#
# From keystone
#

# Name of the paste configuration file that defines the available pipelines.
# (string value)
#config_file = keystone-paste.ini


[policy]

#
# From 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>


[resource]

#
# From keystone
#

# Resource backend driver. If a resource driver is not specified, the
# assignment driver will choose the resource driver. (string value)
#driver = <None>
```

```
# Toggle for resource caching. This has no effect unless global caching is
# enabled. (boolean value)
# Deprecated group/name - [assignment]/caching
#caching = true

# TTL (in seconds) to cache resource data. This has no effect unless global
# caching is enabled. (integer value)
# Deprecated group/name - [assignment]/cache_time
#cache_time = <None>

# Maximum number of entities that will be returned in a resource collection.
# (integer value)
# Deprecated group/name - [assignment]/list_limit
#list_limit = <None>


[revoke]

#
# From keystone
#

# An implementation of the backend for persisting revocation events. (string
# value)
#driver = keystone.contrib.revoke.backends.sql.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 caching. This has no effect unless global
 caching
# is enabled. (boolean value)
#caching = true

# Time to cache the revocation list and the revocation events (in seconds).
# This has no effect unless global and token caching are enabled. (integer
# value)
# Deprecated group/name - [token]/revocation_cache_time
#cache_time = 3600


[role]

#
# From keystone
#

# Role backend driver. (string value)
#driver = <None>

# Toggle for role caching. This has no effect unless global caching is
 enabled.
# (boolean value)
#caching = true

# TTL (in seconds) to cache role data. This has no effect unless global
 caching
# is enabled. (integer value)
```

```
#cache_time = <None>

# Maximum number of entities that will be returned in a role collection.
# (integer value)
#list_limit = <None>


[saml]

#
# From keystone
#

# Default TTL, in seconds, for any generated SAML assertion created by
# Keystone. (integer value)
#assertion_expiration_time = 3600

# Binary to be called for XML signing. Install the appropriate package,
 specify
# absolute path or adjust your PATH environment variable if the binary cannot
# be found. (string value)
#xmlsec1_binary = xmlsec1

# Path of the certfile for SAML signing. For non-production environments, you
# may be interested in using `keystone-manage pki_setup` to generate self-
# signed certificates. Note, the path cannot contain a comma. (string value)
#certfile = /etc/keystone/ssl/certs/signing_cert.pem

# Path of the keyfile for SAML signing. Note, the path cannot contain a comma.
# (string value)
#keyfile = /etc/keystone/ssl/private/signing_key.pem

# Entity ID value for unique Identity Provider identification. Usually FQDN is
# set with a suffix. A value is required to generate IDP Metadata. For
 example:
# https://keystone.example.com/v3/OS-FEDERATION/saml2/idp (string value)
#idp_entity_id = <None>

# Identity Provider Single-Sign-On service value, required in the Identity
# Provider's metadata. A value is required to generate IDP Metadata. For
# example: https://keystone.example.com/v3/OS-FEDERATION/saml2/sso (string
# value)
#idp_sso_endpoint = <None>

# Language used by the organization. (string value)
#idp_lang = en

# Organization name the installation belongs to. (string value)
#idp_organization_name = <None>

# Organization name to be displayed. (string value)
#idp_organization_display_name = <None>

# URL of the organization. (string value)
#idp_organization_url = <None>

# Company of contact person. (string value)
#idp_contact_company = <None>

# Given name of contact person (string value)
```

```
#idp_contact_name = <None>

# Surname of contact person. (string value)
#idp_contact_surname = <None>

# Email address of contact person. (string value)
#idp_contact_email = <None>

# Telephone number of contact person. (string value)
#idp_contact_telephone = <None>

# Contact type. Allowed values are: technical, support, administrative
 billing,
# and other (string value)
#idp_contact_type = other

# Path to the Identity Provider Metadata file. This file should be generated
# with the keystone-manage saml_idp_metadata command. (string value)
#idp_metadata_path = /etc/keystone/saml2_idp_metadata.xml

# The prefix to use for the RelayState SAML attribute, used when generating
 ECP
# wrapped assertions. (string value)
#relay_state_prefix = ss:mem:


[signing]

#
# From keystone
#

# Path of the certfile for token signing. For non-production environments, you
# may be interested in using `keystone-manage pki_setup` to generate self-
# signed certificates. (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

# Days 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]
```

```
#
# From keystone
#

# Path of the CA key file for SSL. (string value)
#ca_key = /etc/keystone/ssl/private/cakey.pem

# SSL key length (in bits) (auto generated certificate). (integer value)
#key_size = 1024

# Days the certificate is valid for once signed (auto generated certificate).
# (integer value)
#valid_days = 3650

# SSL certificate subject (auto generated certificate). (string value)
#cert_subject = /C=US/ST=Unset/L=Unset/O=Unset/CN=localhost


[token]

#
# From keystone
#

# External auth mechanisms that should add bind information to token, e.g.,
# kerberos,x509. (list value)
#bind =

# 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.
# (string value)
#enforce_token_bind = permissive

# Amount of time a token should remain valid (in seconds). (integer value)
#expiration = 3600

# Controls the token construction, validation, and revocation operations. Core
# providers are "keystone.token.providers.[fernet|pkiz|pki|uuid].Provider".
# (string value)
#provider = keystone.token.providers.uuid.Provider

# Token persistence backend driver. (string value)
#driver = keystone.token.persistence.backends.sql.Token

# Toggle for token system caching. This has no effect unless global caching is
# enabled. (boolean value)
#caching = true

# 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

# Allow rescoping of scoped token. Setting allow_rescoped_scoped_token to
 false
# prevents a user from exchanging a scoped token for any other token. (boolean
# value)
#allow_rescope_scoped_token = true

# The hash algorithm to use for PKI tokens. This can be set to any algorithm
# that hashlib supports. WARNING: Before changing this value, the auth_token
# middleware must be configured with the hash_algorithms, otherwise token
# revocation will not be processed correctly. (string value)
#hash_algorithm = md5


[trust]

#
# From keystone
#

# Delegation and impersonation features can be optionally disabled. (boolean
# value)
#enabled = true

# Enable redelegation feature. (boolean value)
#allow_redelegation = false

# Maximum depth of trust redelegation. (integer value)
#max_redelegation_count = 3

# Trust backend driver. (string value)
#driver = keystone.trust.backends.sql.Trust
```

keystone-paste.ini

Use the `keystone-paste.ini` file to configure 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:request_id]
paste.filter_factory = oslo_middleware:RequestId.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: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:endpoint_policy_extension]
paste.filter_factory = keystone.contrib.endpoint_policy.
routers:EndpointPolicyExtension.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 = oslo_middleware.sizelimit:RequestBodySizeLimiter.
factory

[app:public_service]
paste.app_factory = keystone.service:public_app_factory

[app:service_v3]
paste.app_factory = keystone.service:v3_app_factory

[app:admin_service]
paste.app_factory = keystone.service:admin_app_factory

[pipeline:public_api]
# The last item in this pipeline must be public_service or an equivalent
# application. It cannot be a filter.
```

```
pipeline = sizelimit url_normalize request_id build_auth_context token_auth
 admin_token_auth json_body ec2_extension user_crud_extension public_service

[pipeline:admin_api]
# The last item in this pipeline must be admin_service or an equivalent
# application. It cannot be a filter.
pipeline = sizelimit url_normalize request_id build_auth_context token_auth
 admin_token_auth json_body ec2_extension s3_extension crud_extension
 admin_service

[pipeline:api_v3]
# The last item in this pipeline must be service_v3 or an equivalent
# application. It cannot be a filter.
pipeline = sizelimit url_normalize request_id build_auth_context
 token_auth admin_token_auth json_body ec2_extension_v3 s3_extension
 simple_cert_extension revoke_extension federation_extension oauth1_extension
 endpoint_filter_extension endpoint_policy_extension service_v3

[app:public_version_service]
paste.app_factory = keystone.service:public_version_app_factory

[app:admin_version_service]
paste.app_factory = keystone.service:admin_version_app_factory

[pipeline:public_version_api]
pipeline = sizelimit url_normalize public_version_service

[pipeline:admin_version_api]
pipeline = sizelimit url_normalize admin_version_service

[composite:main]
use = egg:Paste#urlmap
/v2.0 = public_api
/v3 = api_v3
/ = public_version_api

[composite:admin]
use = egg:Paste#urlmap
/v2.0 = admin_api
/v3 = api_v3
/ = admin_version_api
```

logging.conf

You can specify a special logging configuration file in the `keystone.conf` configuration file. For example, `/etc/keystone/logging.conf`.

For details, see the (Python logging module documentation).

```
[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,)


##################
# Log Formatters #
##################

[formatter_minimal]
format=%(message)s

[formatter_normal]
format=(%(name)s): %(asctime)s %(levelname)s %(message)s

[formatter_debug]
format=(%(name)s): %(asctime)s %(levelname)s %(module)s %(funcName)s
 %(message)s
```

policy.json

Use the `policy.json` file to define additional access controls that apply to the Identity service.

```
{
    "admin_required": "role:admin or is_admin:1",
    "service_role": "role:service",
    "service_or_admin": "rule:admin_required or rule:service_role",
    "owner" : "user_id:%(user_id)s",
    "admin_or_owner": "rule:admin_required or rule:owner",
    "token_subject": "user_id:%(target.token.user_id)s",
    "admin_or_token_subject": "rule:admin_required or rule:token_subject",

    "default": "rule:admin_required",

    "identity:get_region": "",
    "identity:list_regions": "",
    "identity:create_region": "rule:admin_required",
    "identity:update_region": "rule:admin_required",
    "identity:delete_region": "rule:admin_required",

    "identity:get_service": "rule:admin_required",
    "identity:list_services": "rule:admin_required",
    "identity:create_service": "rule:admin_required",
    "identity:update_service": "rule:admin_required",
    "identity:delete_service": "rule:admin_required",

    "identity:get_endpoint": "rule:admin_required",
    "identity:list_endpoints": "rule:admin_required",
    "identity:create_endpoint": "rule:admin_required",
    "identity:update_endpoint": "rule:admin_required",
    "identity:delete_endpoint": "rule:admin_required",

    "identity:get_domain": "rule:admin_required",
    "identity:list_domains": "rule:admin_required",
    "identity:create_domain": "rule:admin_required",
    "identity:update_domain": "rule:admin_required",
    "identity:delete_domain": "rule:admin_required",

    "identity:get_project": "rule:admin_required",
    "identity:list_projects": "rule:admin_required",
    "identity:list_user_projects": "rule:admin_or_owner",
    "identity:create_project": "rule:admin_required",
    "identity:update_project": "rule:admin_required",
    "identity:delete_project": "rule:admin_required",

    "identity:get_user": "rule:admin_required",
    "identity:list_users": "rule:admin_required",
    "identity:create_user": "rule:admin_required",
    "identity:update_user": "rule:admin_required",
    "identity:delete_user": "rule:admin_required",
    "identity:change_password": "rule:admin_or_owner",

    "identity:get_group": "rule:admin_required",
    "identity:list_groups": "rule:admin_required",
    "identity:list_groups_for_user": "rule:admin_or_owner",
    "identity:create_group": "rule:admin_required",
    "identity:update_group": "rule:admin_required",
```

```
    "identity:delete_group": "rule:admin_required",
    "identity:list_users_in_group": "rule:admin_required",
    "identity:remove_user_from_group": "rule:admin_required",
    "identity:check_user_in_group": "rule:admin_required",
    "identity:add_user_to_group": "rule:admin_required",

    "identity:get_credential": "rule:admin_required",
    "identity:list_credentials": "rule:admin_required",
    "identity:create_credential": "rule:admin_required",
    "identity:update_credential": "rule:admin_required",
    "identity:delete_credential": "rule:admin_required",

    "identity:ec2_get_credential": "rule:admin_required or (rule:owner and
user_id:%(target.credential.user_id)s)",
    "identity:ec2_list_credentials": "rule:admin_or_owner",
    "identity:ec2_create_credential": "rule:admin_or_owner",
    "identity:ec2_delete_credential": "rule:admin_required or (rule:owner and
user_id:%(target.credential.user_id)s)",

    "identity:get_role": "rule:admin_required",
    "identity:list_roles": "rule:admin_required",
    "identity:create_role": "rule:admin_required",
    "identity:update_role": "rule:admin_required",
    "identity:delete_role": "rule:admin_required",

    "identity:check_grant": "rule:admin_required",
    "identity:list_grants": "rule:admin_required",
    "identity:create_grant": "rule:admin_required",
    "identity:revoke_grant": "rule:admin_required",

    "identity:list_role_assignments": "rule:admin_required",

    "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_token_subject",

    "identity:create_trust": "user_id:%(trust.trustor_user_id)s",
    "identity:get_trust": "rule:admin_or_owner",
    "identity:list_trusts": "",
    "identity:list_roles_for_trust": "",
    "identity:get_role_for_trust": "",
    "identity:delete_trust": "",

    "identity:create_consumer": "rule:admin_required",
    "identity:get_consumer": "rule:admin_required",
    "identity:list_consumers": "rule:admin_required",
    "identity:delete_consumer": "rule:admin_required",
    "identity:update_consumer": "rule:admin_required",

    "identity:authorize_request_token": "rule:admin_required",
    "identity:list_access_token_roles": "rule:admin_required",
    "identity:get_access_token_role": "rule:admin_required",
```

```
    "identity:list_access_tokens": "rule:admin_required",
    "identity:get_access_token": "rule:admin_required",
    "identity:delete_access_token": "rule:admin_required",

    "identity:list_projects_for_endpoint": "rule:admin_required",
    "identity:add_endpoint_to_project": "rule:admin_required",
    "identity:check_endpoint_in_project": "rule:admin_required",
    "identity:list_endpoints_for_project": "rule:admin_required",
    "identity:remove_endpoint_from_project": "rule:admin_required",

    "identity:create_endpoint_group": "rule:admin_required",
    "identity:list_endpoint_groups": "rule:admin_required",
    "identity:get_endpoint_group": "rule:admin_required",
    "identity:update_endpoint_group": "rule:admin_required",
    "identity:delete_endpoint_group": "rule:admin_required",
    "identity:list_projects_associated_with_endpoint_group":
"rule:admin_required",
    "identity:list_endpoints_associated_with_endpoint_group":
"rule:admin_required",
    "identity:get_endpoint_group_in_project": "rule:admin_required",
    "identity:add_endpoint_group_to_project": "rule:admin_required",
    "identity:remove_endpoint_group_from_project": "rule:admin_required",

    "identity:create_identity_provider": "rule:admin_required",
    "identity:list_identity_providers": "rule:admin_required",
    "identity:get_identity_providers": "rule:admin_required",
    "identity:update_identity_provider": "rule:admin_required",
    "identity:delete_identity_provider": "rule:admin_required",

    "identity:create_protocol": "rule:admin_required",
    "identity:update_protocol": "rule:admin_required",
    "identity:get_protocol": "rule:admin_required",
    "identity:list_protocols": "rule:admin_required",
    "identity:delete_protocol": "rule:admin_required",

    "identity:create_mapping": "rule:admin_required",
    "identity:get_mapping": "rule:admin_required",
    "identity:list_mappings": "rule:admin_required",
    "identity:delete_mapping": "rule:admin_required",
    "identity:update_mapping": "rule:admin_required",

    "identity:create_service_provider": "rule:admin_required",
    "identity:list_service_providers": "rule:admin_required",
    "identity:get_service_provider": "rule:admin_required",
    "identity:update_service_provider": "rule:admin_required",
    "identity:delete_service_provider": "rule:admin_required",

    "identity:get_auth_catalog": "",
    "identity:get_auth_projects": "",
    "identity:get_auth_domains": "",

    "identity:list_projects_for_groups": "",
    "identity:list_domains_for_groups": "",

    "identity:list_revoke_events": "",

    "identity:create_policy_association_for_endpoint": "rule:admin_required",
    "identity:check_policy_association_for_endpoint": "rule:admin_required",
    "identity:delete_policy_association_for_endpoint": "rule:admin_required",
    "identity:create_policy_association_for_service": "rule:admin_required",
```

```
    "identity:check_policy_association_for_service": "rule:admin_required",
    "identity:delete_policy_association_for_service": "rule:admin_required",
    "identity:create_policy_association_for_region_and_service":
 "rule:admin_required",
    "identity:check_policy_association_for_region_and_service":
 "rule:admin_required",
    "identity:delete_policy_association_for_region_and_service":
 "rule:admin_required",
    "identity:get_policy_for_endpoint": "rule:admin_required",
    "identity:list_endpoints_for_policy": "rule:admin_required",

    "identity:create_domain_config": "rule:admin_required",
    "identity:get_domain_config": "rule:admin_required",
    "identity:update_domain_config": "rule:admin_required",
    "identity:delete_domain_config": "rule:admin_required"
}
```

Domain-specific configuration

Identity enables you to configure domain-specific authentication drivers. For example, you can configure a domain to have its own LDAP or SQL server.

By default, the option to configure domain-specific drivers is disabled.

To enable domain-specific drivers, set these options in `[identity]` section in the `keystone.conf` file:

```
[identity]
domain_specific_drivers_enabled = True
domain_config_dir = /etc/keystone/domains
```

When you enable domain-specific drivers, Identity looks in the `domain_config_dir` directory for configuration files that are named as follows: `keystone.`*`DOMAIN_NAME`*`.conf`, where *`DOMAIN_NAME`* is the domain name.

Any options that you define in the domain-specific configuration file override options in the primary configuration file for the specified domain. Any domain without a domain-specific configuration file uses only the options in the primary configuration file.

New, updated and deprecated options in Kilo for OpenStack Identity

Table 7.38. New options

Option = default value	(Type) Help string
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.

Option = default value	(Type) Help string
[DEFAULT] 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.
[DEFAULT] max_project_tree_depth = 5	(IntOpt) Maximum depth of the project hierarchy. WARNING: setting it to a large value may adversely impact performance.
[DEFAULT] notification_format = basic	(StrOpt) Define the notification format for Identity Service events. A "basic" notification has information about the resource being operated on. A "cadf" notification has the same information, as well as information about the initiator of the event. Valid options are: basic and cadf
[DEFAULT] secure_proxy_ssl_header = None	(StrOpt) The HTTP header used to determine the scheme for the original request, even if it was removed by an SSL terminating proxy. Typical value is "HTTP_X_FORWARDED_PROTO".
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
[auth] oauth1 = keystone.auth.plugins.oauth1.OAuth	(StrOpt) The oAuth1.0 auth plugin module.
[domain_config] cache_time = 300	(IntOpt) TTL (in seconds) to cache domain config data. This has no effect unless domain config caching is enabled.
[domain_config] caching = True	(BoolOpt) Toggle for domain config caching. This has no effect unless global caching is enabled.
[domain_config] driver = keystone.resource.config_backends.sql.DomainConfig	(StrOpt) Domain config backend driver.
[eventlet_server] admin_bind_host = 0.0.0.0	(StrOpt) The IP address of the network interface for the admin service to listen on.
[eventlet_server] admin_port = 35357	(IntOpt) The port number which the admin service listens on.
[eventlet_server] admin_workers = None	(IntOpt) The number of worker processes to serve the admin eventlet application. Defaults to number of CPUs (minimum of 2).
[eventlet_server] public_bind_host = 0.0.0.0	(StrOpt) The IP address of the network interface for the public service to listen on.
[eventlet_server] public_port = 5000	(IntOpt) The port number which the public service listens on.
[eventlet_server] public_workers = None	(IntOpt) The number of worker processes to serve the public eventlet application. Defaults to number of CPUs (minimum of 2).
[eventlet_server] 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.
[eventlet_server] tcp_keepidle = 600	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Only applies if tcp_keepalive is true.
[eventlet_server_ssl] ca_certs = /etc/keystone/ssl/certs/ca.pem	(StrOpt) Path of the CA cert file for SSL.
[eventlet_server_ssl] cert_required = False	(BoolOpt) Require client certificate.

Option = default value	(Type) Help string
[eventlet_server_ssl] certfile = /etc/keystone/ssl/certs/keystone.pem	(StrOpt) Path of the certfile for SSL. For non-production environments, you may be interested in using `keystone-manage ssl_setup` to generate self-signed certificates.
[eventlet_server_ssl] enable = False	(BoolOpt) Toggle for SSL support on the Keystone eventlet servers.
[eventlet_server_ssl] keyfile = /etc/keystone/ssl/private/keystonekey.pem	(StrOpt) Path of the keyfile for SSL.
[federation] federated_domain_name = Federated	(StrOpt) A domain name that is reserved to allow federated ephemeral users to have a domain concept. Note that an admin will not be able to create a domain with this name or update an existing domain to this name. You are not advised to change this value unless you really have to. Changing this option to empty string or None will not have any impact and default name will be used.
[federation] remote_id_attribute = None	(StrOpt) Value to be used to obtain the entity ID of the Identity Provider from the environment (e.g. if using the mod_shib plugin this value is `Shib-Identity-Provider`).
[federation] sso_callback_template = /etc/keystone/sso_callback_template.html	(StrOpt) Location of Single Sign-On callback handler, will return a token to a trusted dashboard host.
[federation] trusted_dashboard = []	(MultiStrOpt) A list of trusted dashboard hosts. Before accepting a Single Sign-On request to return a token, the origin host must be a member of the trusted_dashboard list. This configuration option may be repeated for multiple values. For example: trusted_dashboard=http://acme.com trusted_dashboard=http://beta.com
[fernet_tokens] key_repository = /etc/keystone/fernet-keys/	(StrOpt) Directory containing Fernet token keys.
[fernet_tokens] max_active_keys = 3	(IntOpt) This controls how many keys are held in rotation by keystone-manage fernet_rotate before they are discarded. The default value of 3 means that keystone will maintain one staged key, one primary key, and one secondary key. Increasing this value means that additional secondary keys will be kept in the rotation.
[identity] cache_time = 600	(IntOpt) Time to cache identity data (in seconds). This has no effect unless global and identity caching are enabled.
[identity] caching = True	(BoolOpt) Toggle for identity caching. This has no effect unless global caching is enabled.
[identity] domain_configurations_from_database = False	(BoolOpt) Extract the domain specific configuration options from the resource backend where they have been stored with the domain data. This feature is disabled by default (in which case the domain specific options will be loaded from files in the domain configuration directory); set to true to enable.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients

Option = default value	(Type) Help string
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.
[oslo_policy] policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
[oslo_policy] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[oslo_policy] policy_file = policy.json	(StrOpt) The JSON file that defines policies.
[resource] cache_time = None	(IntOpt) TTL (in seconds) to cache resource data. This has no effect unless global caching is enabled.
[resource] caching = True	(BoolOpt) Toggle for resource caching. This has no effect unless global caching is enabled.
[resource] driver = None	(StrOpt) Resource backend driver. If a resource driver is not specified, the assignment driver will choose the resource driver.
[resource] list_limit = None	(IntOpt) Maximum number of entities that will be returned in a resource collection.
[revoke] cache_time = 3600	(IntOpt) Time to cache the revocation list and the revocation events (in seconds). This has no effect unless global and token caching are enabled.
[role] cache_time = None	(IntOpt) TTL (in seconds) to cache role data. This has no effect unless global caching is enabled.
[role] caching = True	(BoolOpt) Toggle for role caching. This has no effect unless global caching is enabled.
[role] driver = None	(StrOpt) Role backend driver.
[role] list_limit = None	(IntOpt) Maximum number of entities that will be returned in a role collection.
[saml] relay_state_prefix = ss:mem:	(StrOpt) The prefix to use for the RelayState SAML attribute, used when generating ECP wrapped assertions.
[token] allow_rescope_scoped_token = True	(BoolOpt) Allow rescoping of scoped token. Setting allow_rescoped_scoped_token to false prevents a user from exchanging a scoped token for any other token.
[trust] allow_redelegation = False	(BoolOpt) Enable redelegation feature.

Option = default value	(Type) Help string
[trust] max_redelegation_count = 3	(IntOpt) Maximum depth of trust redelegation.

Table 7.39. New default values

Option	Previous default value	New default value
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost
[auth] methods	external, password, token	external, password, token, oauth1
[revoke] driver	keystone.contrib.revoke.backends.kvs.Revoke	keystone.contrib.revoke.backends.sql.Revoke
[token] provider	None	keystone.token.providers.uuid.Provider

Table 7.40. Deprecated options

Deprecated option	New Option
[DEFAULT] admin_bind_host	[eventlet_server] admin_bind_host
[DEFAULT] log-format	None
[DEFAULT] use-syslog	None
[DEFAULT] admin_workers	[eventlet_server] admin_workers
[assignment] list_limit	[resource] list_limit
[DEFAULT] admin_port	[eventlet_server] admin_port
[assignment] caching	[resource] caching
[DEFAULT] max_request_body_size	[oslo_middleware] max_request_body_size
[assignment] cache_time	[resource] cache_time
[DEFAULT] tcp_keepidle	[eventlet_server] tcp_keepidle
[ssl] cert_required	[eventlet_server_ssl] cert_required
[DEFAULT] public_port	[eventlet_server] public_port
[DEFAULT] public_bind_host	[eventlet_server] public_bind_host
[DEFAULT] tcp_keepalive	[eventlet_server] tcp_keepalive
[token] revocation_cache_time	[revoke] cache_time
[DEFAULT] public_workers	[eventlet_server] public_workers
[ssl] keyfile	[eventlet_server_ssl] keyfile
[ssl] ca_certs	[eventlet_server_ssl] ca_certs
[ssl] enable	[eventlet_server_ssl] enable
[ssl] certfile	[eventlet_server_ssl] certfile

8. Image service

Table of Contents

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 3.30, "Description of glance configuration options" [295] and Table 3.51, "Description of S3 configuration options" [308].

You can modify many options in the OpenStack Image service. The following tables provide a comprehensive list.

Table 8.1. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
`admin_password` = *None*	(StrOpt) Service user password.
`admin_tenant_name` = *admin*	(StrOpt) Service tenant name.
`admin_token` = *None*	(StrOpt) 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.
`admin_user` = *None*	(StrOpt) Service username.
`auth_admin_prefix` =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
`auth_host` = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_plugin` = *None*	(StrOpt) Name of the plugin to load
`auth_port` = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_protocol` = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
`auth_section` = *None*	(StrOpt) Config Section from which to load plugin specific options
`auth_uri` = *None*	(StrOpt) Complete public Identity API endpoint.

Configuration option = Default value	Description
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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 8.2. Description of common configuration options

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
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
digest_algorithm = *sha1*	(StrOpt) Digest algorithm which will be used for digital signature; the default is sha1 the default in Kilo for a smooth upgrade process, and it will be updated with sha256 in next release(L). Use the command "openssl list-message-digest-algorithms" to get the available algorithms supported by the version of OpenSSL on the platform. Examples are "sha1", "sha256", "sha512", etc.
host = *localhost*	(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.
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.

Configuration option = Default value	Description
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
listener_workers = *1*	(IntOpt) Number of workers for notification service. A single notification agent is enabled by default.
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
metadata_encryption_key = *None*	(StrOpt) AES key for encrypting store 'location' metadata. This includes, if used, Swift or S3 credentials. Should be set to a random string of length 16, 24 or 32 bytes
metadata_source_path = */etc/glance/metadefs/*	(StrOpt) Path to the directory where json metadata files are stored
property_protection_file = *None*	(StrOpt) The location of the property protection 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_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*	(StrOpt) Set a system wide quota for every user. This value is the total capacity that a user can use across all storage systems. A value of 0 means unlimited.Optional unit can be specified for the value. Accepted units are B, KB, MB, GB and TB representing Bytes, KiloBytes, MegaBytes, GigaBytes and TeraBytes respectively. If no unit is specified then Bytes is assumed. Note that there should not be any space between value and unit and units are case sensitive.
workers = *4*	(IntOpt) The number of child process workers that will be created to service requests. The default will be equal to the number of CPUs available.
[glance_store]	
os_region_name = *None*	(StrOpt) Region name of this node
[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
[keystone_authtoken]	
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.
[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.

Configuration option = Default value	Description
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 = *glance*	(StrOpt) User name to use for OpenStack service access.
[task]	
task_executor = *taskflow*	(StrOpt) Specifies which task executor to be used to run the task scripts.
task_time_to_live = *48*	(IntOpt) Time in hours for which a task lives after, either succeeding or failing
work_dir = *None*	(StrOpt) Work dir for asynchronous task operations. The directory set here will be used to operate over images - normally before they are imported in the destination store. When providing work dir, make sure enough space is provided for concurrent tasks to run efficiently without running out of space. A rough estimation can be done by multiplying the number of `max_workers` - or the N of workers running - by an average image size (e.g 500MB). The image size estimation should be done based on the average size in your deployment. Note that depending on the tasks running you may need to multiply this number by some factor depending on what the task does. For example, you may want to double the available size if image conversion is enabled. All this being said, remember these are just estimations and you should do them based on the worst case scenario and be prepared to act in case they were wrong.

Table 8.3. Description of database configuration options

Configuration option = Default value	Description
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Description
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.
slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = oslo.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 8.4. Description of logging configuration options

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.

Table 8.5. Description of Elasticsearch configuration options

Configuration option = Default value	Description
[elasticsearch]	
hosts = 127.0.0.1:9200	(ListOpt) List of nodes where Elasticsearch instances are running. A single node should be defined as an IP address and port number.

Table 8.6. Description of flagmappings configuration options

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 upper limit (the maximum size of accumulated cache in bytes) beyond which pruner, if running, starts cleaning the images cache.

Configuration option = Default value	Description
image_cache_sqlite_db = *cache.db*	(StrOpt) The path to the sqlite file database that will be used for image cache management.
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 8.7. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.

Configuration option = Default value	Description
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) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
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 will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 8.8. Description of policy configuration options

Configuration option = Default value	Description
[oslo_policy]	
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = *['policy.d']*	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = *policy.json*	(StrOpt) The JSON file that defines policies.

Table 8.9. Description of profiler configuration options

Configuration option = Default value	Description
[profiler]	
enabled = *False*	(BoolOpt) If False fully disable profiling feature.
trace_sqlalchemy = *False*	(BoolOpt) If False doesn't trace SQL requests.

Table 8.10. Description of Redis configuration options

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 8.11. Description of registry configuration options

Configuration option = Default value	Description
[DEFAULT]	
admin_password = *None*	(StrOpt) The administrators password. If "use_user_token" is not in effect, then admin credentials can be specified.
admin_tenant_name = *None*	(StrOpt) The tenant name of the administrative user. If "use_user_token" is not in effect, then admin tenant name can be specified.
admin_user = *None*	(StrOpt) The administrators user name. If "use_user_token" is not in effect, then admin credentials can be specified.
auth_region = *None*	(StrOpt) The region for the authentication service. If "use_user_token" is not in effect and using keystone auth, then region name can be specified.
auth_strategy = *noauth*	(StrOpt) The strategy to use for authentication. If "use_user_token" is not in effect, then auth strategy can be specified.
auth_url = *None*	(StrOpt) The URL to the keystone service. If "use_user_token" is not in effect and using keystone auth, then URL of keystone can be specified.
registry_client_ca_file = *None*	(StrOpt) 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 environment variable to a filepath of the CA cert file.
registry_client_cert_file = *None*	(StrOpt) 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 environment variable to a filepath of the CA cert file
registry_client_insecure = *False*	(BoolOpt) 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.
registry_client_key_file = *None*	(StrOpt) 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 environment variable to a filepath of the key file
registry_client_protocol = *http*	(StrOpt) The protocol to use for communication with the registry server. Either http or https.

Configuration option = Default value	Description
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 8.12. Description of replicator configuration options

Configuration option = Default value	Description
[DEFAULT]	
args = *None*	(ListOpt) Arguments for the command
chunksize = *65536*	(IntOpt) Amount of data to transfer per HTTP write.
command = *None*	(StrOpt) Command to be given to replicator
dontreplicate = *created_at date deleted_at location updated_at*	(StrOpt) List of fields to not replicate.
mastertoken =	(StrOpt) Pass in your authentication token if you have one. This is the token used for the master.
metaonly = *False*	(BoolOpt) Only replicate metadata, not images.
slavetoken =	(StrOpt) Pass in your authentication token if you have one. This is the token used for the slave.
token =	(StrOpt) Pass in your authentication token if you have one. If you use this option the same token is used for both the master and the slave.

Table 8.13. Description of scrubber configuration options

Configuration option = Default value	Description
[DEFAULT]	
wakeup_time = *300*	(IntOpt) Loop time between checking for new items to schedule for delete.

Table 8.14. Description of TaskFlow configuration options

Configuration option = Default value	Description
[taskflow_executor]	
conversion_format = *None*	(StrOpt) The format to which images will be automatically converted. Can be 'qcow2' or 'raw'.
engine_mode = *parallel*	(StrOpt) The mode in which the engine will run. Can be 'serial' or 'parallel'.
max_workers = *10*	(IntOpt) The number of parallel activities executed at the same time by the engine. The value can be greater than one when the engine mode is 'parallel'.

Table 8.15. Description of testing configuration options

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 versions, 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 the options used to configure the APIs, including enabling SSL and modifying WSGI settings are found below.

Table 8.16. Description of API configuration options

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.
`available_plugins` =	(ListOpt) A list of artifacts that are allowed in the format name or name-version. Empty list means that any artifact can be loaded.
`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.
`http_keepalive` = *True*	(BoolOpt) If False, server will return the header "Connection: close", If True, server will return "Connection: Keep-Alive" in its responses. In order to close the client socket connection explicitly after the response is sent and read successfully by the client, you simply have to set this option to False when you create a wsgi server.
`http_timeout` = *600*	(IntOpt) Timeout seconds for HTTP requests. Set it to None to disable timeout.
`image_size_cap` = *1099511627776*	(IntOpt) Maximum size of image a user can upload in bytes. Defaults to 1099511627776 bytes (1 TB).WARNING: this value should only be increased after careful consideration and must be set to a value under 8 EB (9223372036854775808).
`load_enabled` = *True*	(BoolOpt) When false, no artifacts can be loaded regardless of available_plugins. When true, artifacts can be loaded.
`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'.
`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

Configuration option = Default value	Description
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.
public_endpoint = *None*	(StrOpt) Public url to use for versions endpoint. The default is None, which will use the request's host_url attribute to populate the URL base. If Glance is operating behind a proxy, you will want to change this to represent the proxy's URL.
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 keystonemiddleware's 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. 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. Revealing storage location can be a security risk, so use this setting with caution! The overrides show_image_direct_url.
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.
use_user_token = *True*	(BoolOpt) Whether to pass through the user token when making requests to the registry.
[glance_store]	
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 'stores' config option.
store_capabilities_update_min_interval = *0*	(IntOpt) Minimum interval seconds to execute updating dynamic storage capabilities based on backend status then. It's not a periodic routine, the update logic will be executed only when interval seconds elapsed and an operation of store has triggered. The feature will be enabled only when the option value greater then zero.
stores = *file, http*	(ListOpt) List of stores enabled
[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"
[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 '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 8.17. Description of CA and SSL configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
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.

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 8.18. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 8.19. Description of AMQP configuration options

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.
default_publisher_id = *image.localhost*	(StrOpt) Default publisher_id for outgoing notifications.
disabled_notifications =	(ListOpt) List of disabled notifications. A notification can be given either as a notification type to disable a single event, or as a notification group prefix to disable all events within a group. Example: if this config option is set to ["image.create", "metadef_namespace"], then "image.create" notification will not be sent after image is created and none of the notifications for metadefinition namespaces will be sent.
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.

Configuration option = Default value	Description
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 8.20. Description of RPC configuration options

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.
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 8.21. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.

Configuration option = Default value	Description
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
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_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 8.22. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
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'.

Configuration option = Default value	Description
qpid_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Support for ISO images

You can load ISO images into the Image service. You can subsequently boot an ISO image using Compute.

Procedure 8.1. To load an ISO image to an Image service data store

1. In the Image service, run the following command:

```
$ glance image-create --name "ubuntu-14.04.2-server-amd64.iso" \
    --copy-from http://releases.ubuntu.com/14.04.2/ubuntu-14.04.2-server-
amd64.iso \
    --is-public True --container-format bare --disk-format iso
```

   In this command, `ubuntu.iso` is the name for the ISO image after it is loaded to the Image service, and *`ubuntu-14.04.2-server-amd64.iso`* is the name of the source ISO image.

2. Optionally, to confirm the upload in Image Service (glance), run this command:

   Run this command:

```
$ glance image-list
```

Configure back ends

The Image service supports several back ends for storing virtual machine images:

- OpenStack Block Storage (cinder)
- A directory on a local file system
- GridFS
- Ceph RBD
- Amazon S3
- Sheepdog
- OpenStack Object Storage (swift)

- VMware ESX

The following tables detail the options available for each.

Table 8.23. Description of cinder configuration options

Configuration option = Default value	Description
[glance_store]	
`cinder_api_insecure = False`	(BoolOpt) Allow to perform insecure SSL requests to cinder
`cinder_ca_certificates_file = None`	(StrOpt) Location of ca certicates 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_endpoint_template = None`	(StrOpt) Override service catalog lookup with template for cinder endpoint e.g. http://localhost:8776/v1/%(project_id)s
`cinder_http_retries = 3`	(IntOpt) Number of cinderclient retries on failed http calls

Table 8.24. Description of filesystem configuration options

Configuration option = Default value	Description
[glance_store]	
`filesystem_store_datadir = None`	(StrOpt) Directory to which the Filesystem backend store writes images.
`filesystem_store_datadirs = None`	(MultiStrOpt) List of directories and its priorities to which the Filesystem backend store writes images.
`filesystem_store_file_perm = 0`	(IntOpt) The required permission for created image file. In this way the user other service used, e.g. Nova, who consumes the image could be the exclusive member of the group that owns the files created. Assigning it less then or equal to zero means don't change the default permission of the file. This value will be decoded as an octal digit.
`filesystem_store_metadata_file = None`	(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 object. The object should contain the keys 'id' and 'mountpoint'. The value for both keys should be 'string'.

Table 8.25. Description of GridFS configuration options

Configuration option = Default value	Description
[glance_store]	
`mongodb_store_db = None`	(StrOpt) Database to use
`mongodb_store_uri = None`	(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 8.26. Description of RBD configuration options

Configuration option = Default value	Description
[glance_store]	
`rbd_store_ceph_conf = /etc/ceph/ceph.conf`	(StrOpt) Ceph configuration file path. If <None>, librados will locate the default config. If using cephx authen-

Configuration option = Default value	Description
	tication, this file should include a reference to the right keyring in a client.<USER> section
rbd_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.
rbd_store_pool = *localhost*	(StrOpt) RADOS pool in which images are stored.
rbd_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 8.27. Description of S3 configuration options

Configuration option = Default value	Description
[glance_store]	
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_large_object_chunk_size = *10*	(IntOpt) What multipart upload part size, in MB, should S3 use when uploading parts. The size must be greater than or equal to 5M.
s3_store_large_object_size = *100*	(IntOpt) What size, in MB, should S3 start chunking image files and do a multipart upload in S3.
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.
s3_store_thread_pools = *10*	(IntOpt) The number of thread pools to perform a multipart upload in S3.

Table 8.28. Description of Sheepdog configuration options

Configuration option = Default value	Description
[glance_store]	
sheepdog_store_address = *localhost*	(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 8.29. Description of swift configuration options

Configuration option = Default value	Description
[DEFAULT]	
default_swift_reference = *ref1*	(StrOpt) The reference to the default swift account/backing store parameters to use for adding new images.
swift_store_auth_address = *None*	(StrOpt) The address where the Swift authentication service is listening.(deprecated)
swift_store_config_file = *None*	(StrOpt) The config file that has the swift account(s)configs.

Configuration option = Default value	Description
swift_store_key = *None*	(StrOpt) Auth key for the user authenticating against the Swift authentication service. (deprecated)
swift_store_user = *None*	(StrOpt) The user to authenticate against the Swift authentication service (deprecated)
[glance_store]	
default_swift_reference = *ref1*	(StrOpt) The reference to the default swift account/backing store parameters to use for adding new images.
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.(deprecated)
swift_store_auth_insecure = *False*	(BoolOpt) If True, swiftclient won't check for a valid SSL certificate when authenticating.
swift_store_auth_version = *2*	(StrOpt) Version of the authentication service to use. Valid versions are 2 for keystone and 1 for swauth and rackspace. (deprecated)
swift_store_cacert = *None*	(StrOpt) A string giving the CA certificate file to use in SSL connections for verifying certs.
swift_store_config_file = *None*	(StrOpt) The config file that has the swift account(s)configs.
swift_store_container = *glance*	(StrOpt) Container within the account that the account should use for storing images in Swift when using single container mode. In multiple container mode, this will be the prefix for all containers.
swift_store_create_container_on_put = *False*	(BoolOpt) A boolean value that determines if we create the container if it does not exist.
swift_store_endpoint = *None*	(StrOpt) If set, the configured endpoint will be used. If None, the storage url from the auth response will be used.
swift_store_endpoint_type = *publicURL*	(StrOpt) A string giving the endpoint type of the swift service to use (publicURL, adminURL or internalURL). This setting is only used if swift_store_auth_version is 2.
swift_store_key = *None*	(StrOpt) Auth key for the user authenticating against the Swift authentication service. (deprecated)
swift_store_large_object_chunk_size = *200*	(IntOpt) The amount of data written to a temporary disk buffer during the process of chunking the image file.
swift_store_large_object_size = *5120*	(IntOpt) The size, in MB, that Glance will start chunking image files and do a large object manifest in Swift.
swift_store_multi_tenant = *False*	(BoolOpt) If set to True, enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts.
swift_store_multiple_containers_seed = *0*	(IntOpt) When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.Used only when swift_store_multi_tenant is disabled. The total number of containers that will be used is equal to 16^N, so if this config option is set to 2, then 16^2=256 containers will be used to store images.
swift_store_region = *None*	(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.
swift_store_retry_get_count = *0*	(IntOpt) The number of times a Swift download will be retried before the request fails.

Configuration option = Default value	Description
swift_store_service_type = *object-store*	(StrOpt) A string giving the service type of the swift service to use. This setting is only used if swift_store_auth_version is 2.
swift_store_ssl_compression = *True*	(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.
swift_store_user = *None*	(StrOpt) The user to authenticate against the Swift authentication service (deprecated)

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" [479].

Note

If you intend to use multiple data stores for the back end, use the SPBM feature.

In the `glance_store` section, set the `default_store` option to **`vsphere`**, as shown in this code sample:

```
[glance_store]
# 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 8.30. Description of VMware configuration options

Configuration option = Default value	Description
[glance_store]	
vmware_api_insecure = *False*	(BoolOpt) Allow to perform insecure SSL requests to ESX/VC.

Configuration option = Default value	Description
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) DEPRECATED. 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". This option is deprecated in favor of vmware_datastores and will be removed in the Liberty release.
vmware_datastore_name = *None*	(StrOpt) DEPRECATED. Datastore associated with the datacenter. This option is deprecated in favor of vmware_datastores and will be removed in the Liberty release.
vmware_datastores = *None*	(MultiStrOpt) A list of datastores where the image can be stored. This option may be specified multiple times for specifying multiple datastores. Either one of vmware_datastore_name or vmware_datastores is required. The datastore name should be specified after its datacenter path, seperated by ":". An optional weight may be given after the datastore name, seperated again by ":". Thus, the required format becomes <datacenter_path>:<datastore_name>:<optional_weight>. When adding an image, the datastore with highest weight will be selected, unless there is not enough free space available in cases where the image size is already known. If no weight is given, it is assumed to be zero and the directory will be considered for selection last. If multiple datastores have the same weight, then the one with the most free space available is selected.
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/wsdl/pbmService.wsdl Not setting this
# will disable storage policy based placement of images.
# (string value)
#vmware_pbm_wsdl_location =

# The PBM policy. If `pbm_wsdl_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.

Procedure 8.2. To configure a single data store

1. If present, comment or delete the `vmware_pbm_wsdl_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.

Image service sample configuration files

You can find the files that are described in this section 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
```

```
# 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_KEEPIDLE value in seconds when creating socket.
# Not supported on OS X.
#tcp_keepidle = 600

# Timeout (in seconds) for client connections' socket operations. If an
 incoming
# connection is idle for this period it will be closed.  A value of "0"
# means wait forever.
#client_socket_timeout = 0

# 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

# The number of child process workers that will be
# created to service API requests. The default will be
# equal to the number of CPUs available. (integer value)
#workers = 4

# 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

# 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

# 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


# Public url to use for versions endpoint. The default is None,
# which will use the request's host_url attribute to populate the URL base.
# If Glance is operating behind a proxy, you will want to change this to
# represent the proxy's URL.
#public_endpoint=<None>

# http_keepalive option. If False, server will return the header
# "Connection: close", If True, server will return "Connection: Keep-Alive"
# in its responses. In order to close the client socket connection
# explicitly after the response is sent and read successfully by the client,
# you simply have to set this option to False when you create a wsgi server.
#http_keepalive = True

# ================ 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>


# Digest algorithm which will be used for digital signature, the default is
# sha1 in Kilo for a smooth upgrade process, and it will be updated with
# sha256 in next release(L). Use command
# "openssl list-message-digest-algorithms" to get the available algorithms
# supported by the version of OpenSSL on the platform. Examples are 'sha1',
# 'sha256', 'sha512', etc.
#digest_algorithm = sha1

# ============ 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.
# Default: 600
#registry_client_timeout = 600

# Enable DEBUG log messages from sqlalchemy which prints every database
# query and response.
# Default: False
#sqlalchemy_debug = True

# Pass the user's token through for API requests to the registry.
# Default: True
#use_user_token = True

# If 'use_user_token' is not in effect then admin credentials
# can be specified. Requests to the registry on behalf of
# the API will use these credentials.
# Admin user name
#admin_user = None
# Admin password
#admin_password = None
# Admin tenant name
#admin_tenant_name = None
# Keystone endpoint
#auth_url = None
# Keystone region
#auth_region = None
# Auth strategy
#auth_strategy = keystone

# ============ Notification System Options =====================

# Driver or drivers to handle sending notifications. Set to
# 'messaging' to send notifications to a message queue.
# notification_driver = noop

# Default publisher_id for outgoing notifications.
# default_publisher_id = image.localhost

# List of disabled notifications. A notification can be given either as a
# notification type to disable a single event, or as a notification group
# prefix to disable all events within a group.
# Example: if this config option is set to
# ["image.create", "metadef_namespace"], then "image.create" notification will
# not be sent after image is created and none of the notifications for
# metadefinition namespaces will be sent.
# disabled_notifications = []

# Messaging driver used for 'messaging' notifications driver
# rpc_backend = 'rabbit'

# Configuration options if sending notifications via rabbitmq (these are
# the defaults)
rabbit_host = localhost
rabbit_port = 5672
rabbit_use_ssl = false
rabbit_userid = guest
```

```
rabbit_password = guest
rabbit_virtual_host = /
rabbit_notification_exchange = glance
rabbit_notification_topic = notifications
rabbit_durable_queues = False

# 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

# ============ 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_datadir = /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/
```

```
# ============== Policy Options ===================================

[oslo_policy]
# The JSON file that defines policies.
# Deprecated group/name - [DEFAULT]/policy_file
#policy_file = policy.json

# Default rule. Enforced when a requested rule is not found.
# Deprecated group/name - [DEFAULT]/policy_default_rule
#policy_default_rule = default

# Directories where policy configuration files are stored.
# They can be relative to any directory in the search path
# defined by the config_dir option, or absolute paths.
# The file defined by policy_file must exist for these
# directories to be searched.
# Deprecated group/name - [DEFAULT]/policy_dirs
#policy_dirs = policy.d

# ============== Database Options =================================

[database]
# The file name to use with SQLite (string value)
#sqlite_db = oslo.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,
# 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

[oslo_concurrency]
```

```
# Enables or disables inter-process locks. (boolean value)
# Deprecated group/name - [DEFAULT]/disable_process_locking
#disable_process_locking = false

# Directory to use for lock files.  For security, the specified
# directory should only be writable by the user running the processes
# that need locking. It could be read from environment variable
# OSLO_LOCK_PATH. This setting needs to be the same for both
# glance-scrubber and glance-api service. Default to a temp directory.
# Deprecated group/name - [DEFAULT]/lock_path (string value)
#lock_path = /tmp

[keystone_authtoken]
identity_uri = http://127.0.0.1:35357
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%
revocation_cache_time = 10

[paste_deploy]
# Name of the paste configuration file that defines the available pipelines
#config_file = glance-api-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-api-keystone], you would configure the flavor below
# as 'keystone'.
#flavor=

[store_type_location_strategy]
# The scheme list to use to get store preference order. The scheme must be
# registered by one of the stores defined by the '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.
#store_type_preference =

[profiler]
# If False fully disable profiling feature.
#enabled = False

# If False doesn't trace SQL requests.
#trace_sqlalchemy = False

[task]
# ================= Glance Tasks Options ============================

# 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

# Specifies which task executor to be used to run the task scripts.
# The default value for task_executor is taskflow.
# task_executor = taskflow

# Work dir for asynchronous task operations.  The directory set here
# will be used to operate over images - normally before they are
# imported in the destination store. When providing work dir, make sure
# enough space is provided for concurrent tasks to run efficiently
# without running out of space. A rough estimation can be done by
```

```
# multiplying the number of `max_workers` - or the N of workers running
# - by an average image size (e.g 500MB). The image size estimation
# should be done based on the average size in your deployment. Note that
# depending on the tasks running you may need to multiply this number by
# some factor depending on what the task does. For example, you may want
# to double the available size if image conversion is enabled. All this
# being said, remember these are just estimations and you should do them
# based on the worst case scenario and be prepared to act in case they
# were wrong.
# work_dir=None

# Specifies the maximum number of eventlet threads which can be spun up by
# the eventlet based task executor to perform execution of Glance tasks.
# DEPRECATED: Use [taskflow_executor]/max_workers instead.
# eventlet_executor_pool_size = 1000

[taskflow_executor]
# The mode in which the engine will run. Can be 'default', 'serial',
# 'parallel' or 'worker-based'
#engine_mode = serial

# The number of parallel activities executed at the same time by
# the engine. The value can be greater than one when the engine mode is
# 'parallel' or 'worker-based', otherwise this value will be ignored.
#max_workers = 10

[glance_store]
# List of which store classes and store class locations are
# currently known to glance at startup.
# Deprecated group/name - [DEFAULT]/known_stores
# 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,
#stores = glance.store.filesystem.Store,
#         glance.store.http.Store

# 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 stores option.
# Deprecated group/name - [DEFAULT]/default_store
# Default: 'file'
default_store = file

# ============ 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 rackspace
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

# swift_store_config_file = glance-swift.conf
# This file contains references for each of the configured
# Swift accounts/backing stores. If used, this option can prevent
# credentials being stored in the database. Using Swift references
# is disabled if this config is left blank.

# The reference to the default Swift parameters to use for adding new images.
# default_swift_reference = 'ref1'

# 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

# If set, the configured endpoint will be used. If None, the storage URL
# from the auth response will be used. The location of an object is
# obtained by appending the container and object to the configured URL.
#
# swift_store_endpoint = https://www.example.com/v1/not_a_container
#swift_store_endpoint =

# 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

# If set to an integer value between 1 and 32, a single-tenant store will
# use multiple containers to store images. If set to the default value of 0,
# only a single container will be used. Multi-tenant stores are not affected
# by this option. The max number of containers that will be used to store
# images is approximately 16^N where N is the value of this option. Discuss
# the impact of this with your swift deployment team, as this option is only
# beneficial in the largest of deployments where swift rate limiting can lead
# to unwanted throttling on a single container.
#swift_store_multiple_containers_seed = 0

# 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

# The number of times a Swift download will be retried before the
# request fails
#swift_store_retry_get_count = 0

# Bypass SSL verification for Swift
#swift_store_auth_insecure = False

# The path to a CA certificate bundle file to use for SSL verification when
# communicating with Swift.
#swift_store_cacert =

# ============ S3 Store Options =============================
```

```
# Address where the S3 authentication service lives
# Valid schemes are 'http://' and 'https://'
# If no scheme specified,  default to 'http://'
s3_store_host = s3.amazonaws.com

# User to authenticate against the S3 authentication service
s3_store_access_key = <20-char AWS access key>

# Auth key for the user authenticating against the
# S3 authentication service
s3_store_secret_key = <40-char AWS secret key>

# Container within the account that the account should use
# for storing images in S3. Note that S3 has a flat namespace,
# so you need a unique bucket name for your glance images. An
# easy way to do this is append your AWS access key to "glance".
# S3 buckets in AWS *must* be lowercased, so remember to lowercase
# your AWS access key if you use it in your bucket name below!
s3_store_bucket = <lowercased 20-char aws access key>glance

# Do we create the bucket if it does not exist?
s3_store_create_bucket_on_put = False

# When sending images to S3, the data will first be written to a
# temporary buffer on disk. By default the platform's temporary directory
# will be used. If required, an alternative directory can be specified here.
#s3_store_object_buffer_dir = /path/to/dir

# When forming a bucket url, boto will either set the bucket name as the
# subdomain or as the first token of the path. Amazon's S3 service will
# accept it as the subdomain, but Swift's S3 middleware requires it be
# in the path. Set this to 'path' or 'subdomain' - defaults to 'subdomain'.
#s3_store_bucket_url_format = subdomain

# Size, in MB, should S3 start chunking image files
# and do a multipart upload in S3. The default is 100MB.
#s3_store_large_object_size = 100

# Multipart upload part size, in MB, should S3 use when uploading
# parts. The size must be greater than or equal to
# 5MB. The default is 10MB.
#s3_store_large_object_chunk_size = 10

# The number of thread pools to perform a multipart upload
# in S3. The default is 10.
#s3_store_thread_pools = 10

# ============ RBD Store Options =============================

# Ceph configuration file path
# If using cephx authentication, this file should
# include a reference to the right keyring
# in a client.<USER> section
#rbd_store_ceph_conf = /etc/ceph/ceph.conf

# 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
#rbd_store_user = <None>
```

```
# RADOS pool in which images are stored
#rbd_store_pool = images

# RADOS images will be chunked into objects of this size (in megabytes).
# For best performance, this should be a power of two
#rbd_store_chunk_size = 8

# ============ Sheepdog Store Options =============================

sheepdog_store_address = localhost

sheepdog_store_port = 7000

# Images will be chunked into objects of this size (in megabytes).
# For best performance, this should be a power of two
sheepdog_store_chunk_size = 64

# ============ Cinder Store Options ===============================

# Info to match when looking for cinder in the service catalog
# Format is : separated values of the form:
# <service_type>:<service_name>:<endpoint_type> (string value)
#cinder_catalog_info = volume:cinder:publicURL

# Override service catalog lookup with template for cinder endpoint
# e.g. http://localhost:8776/v1/%(project_id)s (string value)
#cinder_endpoint_template = <None>

# Region name of this node (string value)
#os_region_name = <None>

# Location of ca certicates file to use for cinder client requests
# (string value)
#cinder_ca_certificates_file = <None>

# Number of cinderclient retries on failed http calls (integer value)
#cinder_http_retries = 3

# Allow to perform insecure SSL requests to cinder (boolean value)
#cinder_api_insecure = False

# ============ 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 = <None>

# Server username (string value)
#vmware_server_username = <None>

# Server password (string value)
#vmware_server_password = <None>

# Inventory path to a datacenter (string value)
# Value optional when vmware_server_ip is an ESX/ESXi host: if specified
# should be `ha-datacenter`.
# Deprecated in favor of vmware_datastores.
#vmware_datacenter_path = <None>
```

```
# Datastore associated with the datacenter (string value)
# Deprecated in favor of vmware_datastores.
#vmware_datastore_name = <None>

# A list of datastores where the image can be stored.
# This option may be specified multiple times for specifying multiple
# datastores. Either one of vmware_datastore_name or vmware_datastores is
# required. The datastore name should be specified after its datacenter
# path, separated by ":". An optional weight may be given after the datastore
# name, separated again by ":". Thus, the required format becomes
# <datacenter_path>:<datastore_name>:<optional_weight>.
# When adding an image, the datastore with highest weight will be selected,
# unless there is not enough free space available in cases where the image
 size
# is already known. If no weight is given, it is assumed to be zero and the
# directory will be considered for selection last. If multiple datastores have
# the same weight, then the one with the most free space available is
 selected.
#vmware_datastores = <None>

# The number of times we retry on failures
# e.g., socket error, etc (integer value)
#vmware_api_retry_count = 10

# 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
```

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

# Timeout (in seconds) for client connections' socket operations. If an
 incoming
# connection is idle for this period it will be closed.  A value of "0"
# means wait forever.
#client_socket_timeout = 0

# API to use for accessing data. Default value points to sqlalchemy
# package.
#data_api = glance.db.sqlalchemy.api

# The number of child process workers that will be
# created to service Registry requests. The default will be
# equal to the number of CPUs available. (integer value)
#workers = None

# 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

# Enable DEBUG log messages from sqlalchemy which prints every database
# query and response.
# Default: False
#sqlalchemy_debug = True

# http_keepalive option. If False, server will return the header
# "Connection: close", If True, server will return "Connection: Keep-Alive"
# in its responses. In order to close the client socket connection
# explicitly after the response is sent and read successfully by the client,
# you simply have to set this option to False when you create a wsgi server.
#http_keepalive = True

# ================ 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_LOCAL1

# ================= SSL Options ===============================

# Certificate file to use when starting registry server securely
#cert_file = /path/to/certfile

# Private key file to use when starting registry server securely
#key_file = /path/to/keyfile

# CA certificate file to use to verify connecting clients
#ca_file = /path/to/cafile

# ============ Notification System Options =====================

# Driver or drivers to handle sending notifications. Set to
# 'messaging' to send notifications to a message queue.
# notification_driver = noop

# Default publisher_id for outgoing notifications.
# default_publisher_id = image.localhost

# Messaging driver used for 'messaging' notifications driver
# rpc_backend = 'rabbit'

# Configuration options if sending notifications via rabbitmq (these are
# the defaults)
rabbit_host = localhost
rabbit_port = 5672
rabbit_use_ssl = false
rabbit_userid = guest
rabbit_password = guest
rabbit_virtual_host = /
rabbit_notification_exchange = glance
rabbit_notification_topic = notifications
rabbit_durable_queues = False

# 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


# ============== Policy Options ================================

[oslo_policy]
```

```
# The JSON file that defines policies.
# Deprecated group/name - [DEFAULT]/policy_file
#policy_file = policy.json

# Default rule. Enforced when a requested rule is not found.
# Deprecated group/name - [DEFAULT]/policy_default_rule
#policy_default_rule = default

# Directories where policy configuration files are stored.
# They can be relative to any directory in the search path
# defined by the config_dir option, or absolute paths.
# The file defined by policy_file must exist for these
# directories to be searched.
# Deprecated group/name - [DEFAULT]/policy_dirs
#policy_dirs = policy.d

# ================= 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,
# 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_authtoken]
identity_uri = http://127.0.0.1:35357
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=

[profiler]
# If False fully disable profiling feature.
#enabled = False

# If False doesn't trace SQL requests.
#trace_sqlalchemy = False
```

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 osprofiler unauthenticated-context rootapp

# Use this pipeline for image caching and no auth
[pipeline:glance-api-caching]
pipeline = versionnegotiation osprofiler unauthenticated-context cache rootapp

# Use this pipeline for caching w/ management interface but no auth
[pipeline:glance-api-cachemanagement]
pipeline = versionnegotiation osprofiler unauthenticated-context cache
 cachemanage rootapp

# Use this pipeline for keystone auth
[pipeline:glance-api-keystone]
pipeline = versionnegotiation osprofiler authtoken context  rootapp

# Use this pipeline for keystone auth with image caching
[pipeline:glance-api-keystone+caching]
pipeline = versionnegotiation osprofiler authtoken context cache rootapp

# Use this pipeline for keystone auth with caching and cache management
[pipeline:glance-api-keystone+cachemanagement]
pipeline = versionnegotiation osprofiler authtoken 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 osprofiler 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 osprofiler 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 = keystonemiddleware.auth_token:filter_factory
delay_auth_decision = true

[filter:gzip]
paste.filter_factory = glance.api.middleware.gzip:GzipMiddleware.factory

[filter:osprofiler]
paste.filter_factory = osprofiler.web:WsgiMiddleware.factory
hmac_keys = SECRET_KEY
enabled = yes
```

glance-manage.conf

The Image service's custom logging options are found in the `glance-manage.conf` file.

Note

Options set in `glance-manage.conf` will override options of the same section and name set in `glance-registry.conf` and `glance-api.conf`. Similarly, options in `glance-api.conf` will override options set in `glance-registry.conf`.

```
[DEFAULT]

#
# From glance.manage
#

# Print debugging output (set logging level to DEBUG instead of
# default WARNING level). (boolean value)
#debug = false

# The name of a logging configuration file. This file is appended to
# any existing logging configuration files. For details about logging
# configuration files, see the Python logging module documentation.
# (string value)
# Deprecated group/name - [DEFAULT]/log_config
#log_config_append = <None>

# Format string for %%(asctime)s in log records. Default: %(default)s
# . (string value)
#log_date_format = %Y-%m-%d %H:%M:%S

# (Optional) The base directory used for relative --log-file paths.
# (string value)
# Deprecated group/name - [DEFAULT]/logdir
#log_dir = <None>

# (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 = /var/log/glance/manage.log

# 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>

# Syslog facility to receive log lines. (string value)
#syslog_log_facility = LOG_USER

# Use syslog for logging. Existing syslog format is DEPRECATED during
# I, and will change in J to honor RFC5424. (boolean value)
#use_syslog = false

# (Optional) Enables or disables syslog rfc5424 format for logging. If
# enabled, prefixes the MSG part of the syslog message with APP-NAME
# (RFC5424). The format without the APP-NAME is deprecated in I, and
# will be removed in J. (boolean value)
#use_syslog_rfc_format = false

# Print more verbose output (set logging level to INFO instead of
# default WARNING level). (boolean value)
#verbose = false


[database]

#
# From oslo.db
#
```

```
# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use 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>

# 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 True, increases the interval between database connection retries
# up to db_max_retry_interval. (boolean value)
#db_inc_retry_interval = true

# Maximum database connection retries before error is raised. Set to
# -1 to specify an infinite retry count. (integer value)
#db_max_retries = 20

# If db_inc_retry_interval is set, the maximum seconds between
# database connection retries. (integer value)
#db_max_retry_interval = 10

# Seconds between database connection retries. (integer value)
#db_retry_interval = 1

# 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

# 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>

# 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 number of database connection retries during startup. Set to
# -1 to specify an infinite retry count. (integer value)
# Deprecated group/name - [DEFAULT]/sql_max_retries
# Deprecated group/name - [DATABASE]/sql_max_retries
#max_retries = 10
```

```
# 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

# 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

# If set, use this value for pool_timeout with SQLAlchemy. (integer
# value)
# Deprecated group/name - [DATABASE]/sqlalchemy_pool_timeout
#pool_timeout = <None>

# 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

# The SQLAlchemy connection string to use to connect to the slave
# database. (string value)
#slave_connection = <None>

# The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
#sqlite_db = oslo.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

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

#
# From oslo.db.concurrency
#

# Enable the experimental use of thread pooling for all DB API calls
# (boolean value)
# Deprecated group/name - [DEFAULT]/dbapi_use_tpool
#use_tpool = false
```

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 = osprofiler unauthenticated-context registryapp

# Use this pipeline for keystone auth
[pipeline:glance-registry-keystone]
```

```
pipeline = osprofiler 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 = osprofiler 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 = keystonemiddleware.auth_token:filter_factory

[filter:osprofiler]
paste.filter_factory = osprofiler.web:WsgiMiddleware.factory
hmac_keys = SECRET_KEY
enabled = yes
```

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
cleanup_scrubber_time = 86400

# Address to find the registry server for cleanups
registry_host = 0.0.0.0

# Port the registry server is listening on
registry_port = 9191

# Auth settings if using Keystone
# auth_url = http://127.0.0.1:5000/v2.0/
# admin_tenant_name = %SERVICE_TENANT_NAME%
# admin_user = %SERVICE_USER%
# admin_password = %SERVICE_PASSWORD%

# 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

# ================= 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>

# ============== Policy Options ================================

# The JSON file that defines policies.
#policy_file = policy.json

# Default rule. Enforced when a requested rule is not found.
#policy_default_rule = default

# Directories where policy configuration files are stored.
# They can be relative to any directory in the search path
# defined by the config_dir option, or absolute paths.
# The file defined by policy_file must exist for these
# directories to be searched.
#policy_dirs = policy.d

# ================= Database Options ===============+==========

[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
#connection=sqlite:////glance/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)
#idle_timeout=3600

# Minimum number of SQL connections to keep open in a pool
# (integer value)
#min_pool_size=1

# Maximum number of SQL connections to keep open in a pool
# (integer value)
#max_pool_size=<None>

# maximum db connection retries during startup. (setting -1
# implies an infinite retry count) (integer value)
#max_retries=10

# interval between retries of opening a sql connection
# (integer value)
#retry_interval=10

# If set, use this value for max_overflow with sqlalchemy
# (integer value)
#max_overflow=<None>

# Verbosity of SQL debugging information. 0=None,
# 100=Everything (integer value)
#connection_debug=0

# Add python stack traces to SQL as comment strings (boolean
# value)
#connection_trace=false

# If set, use this value for pool_timeout with sqlalchemy
# (integer value)
#pool_timeout=<None>

[oslo_concurrency]

# Enables or disables inter-process locks. (boolean value)
# Deprecated group/name - [DEFAULT]/disable_process_locking
#disable_process_locking = false

# Directory to use for lock files.  For security, the specified
# directory should only be writable by the user running the processes
# that need locking. It could be read from environment variable
# OSLO_LOCK_PATH. This setting needs to be the same for both
# glance-scrubber and glance-api service. Default to a temp directory.
# Deprecated group/name - [DEFAULT]/lock_path (string value)
#lock_path = /tmp
```

policy.json

The `/etc/glance/policy.json` file defines additional access controls that apply to the Image service.

```
{
    "context_is_admin":  "role:admin",
    "default": "",
```

```
    "add_image": "",
    "delete_image": "",
    "get_image": "",
    "get_images": "",
    "modify_image": "",
    "publicize_image": "role:admin",
    "copy_from": "",

    "download_image": "",
    "upload_image": "",

    "delete_image_location": "",
    "get_image_location": "",
    "set_image_location": "",

    "add_member": "",
    "delete_member": "",
    "get_member": "",
    "get_members": "",
    "modify_member": "",

    "manage_image_cache": "role:admin",

    "get_task": "",
    "get_tasks": "",
    "add_task": "",
    "modify_task": "",

    "deactivate": "",
    "reactivate": "",

    "get_metadef_namespace": "",
    "get_metadef_namespaces":"",
    "modify_metadef_namespace":"",
    "add_metadef_namespace":"",

    "get_metadef_object":"",
    "get_metadef_objects":"",
    "modify_metadef_object":"",
    "add_metadef_object":"",

    "list_metadef_resource_types":"",
    "get_metadef_resource_type":"",
    "add_metadef_resource_type_association":"",

    "get_metadef_property":"",
    "get_metadef_properties":"",
    "modify_metadef_property":"",
    "add_metadef_property":"",

    "get_metadef_tag":"",
    "get_metadef_tags":"",
    "modify_metadef_tag":"",
    "add_metadef_tag":"",
    "add_metadef_tags":""

}
```

New, updated and deprecated options in Kilo for OpenStack Image service

Table 8.31. New options

Option = default value	(Type) Help string
[DEFAULT] args = None	(ListOpt) Arguments for the command
[DEFAULT] available_plugins =	(ListOpt) A list of artifacts that are allowed in the format name or name-version. Empty list means that any artifact can be loaded.
[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.
[DEFAULT] chunksize = 65536	(IntOpt) Amount of data to transfer per HTTP write.
[DEFAULT] command = None	(StrOpt) Command to be given to replicator
[DEFAULT] digest_algorithm = sha1	(StrOpt) Digest algorithm which will be used for digital signature; the default is sha1 the default in Kilo for a smooth upgrade process, and it will be updated with sha256 in next release(L). Use the command "openssl list-message-digest-algorithms" to get the available algorithms supported by the version of OpenSSL on the platform. Examples are "sha1", "sha256", "sha512", etc.
[DEFAULT] disabled_notifications =	(ListOpt) List of disabled notifications. A notification can be given either as a notification type to disable a single event, or as a notification group prefix to disable all events within a group. Example: if this config option is set to ["image.create", "metadef_namespace"], then "image.create" notification will not be sent after image is created and none of the notifications for metadefinition namespaces will be sent.
[DEFAULT] dontreplicate = created_at date deleted_at location updated_at	(StrOpt) List of fields to not replicate.
[DEFAULT] host = localhost	(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.
[DEFAULT] http_keepalive = True	(BoolOpt) If False, server will return the header "Connection: close", If True, server will return "Connection: Keep-Alive" in its responses. In order to close the client socket connection explicitly after the response is sent and read successfully by the client, you simply have to set this option to False when you create a wsgi server.
[DEFAULT] http_timeout = 600	(IntOpt) Timeout seconds for HTTP requests. Set it to None to disable timeout.
[DEFAULT] listener_workers = 1	(IntOpt) Number of workers for notification service. A single notification agent is enabled by default.
[DEFAULT] load_enabled = True	(BoolOpt) When false, no artifacts can be loaded regardless of available_plugins. When true, artifacts can be loaded.
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.

Option = default value	(Type) Help string
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] mastertoken =	(StrOpt) Pass in your authentication token if you have one. This is the token used for the master.
[DEFAULT] metaonly = False	(BoolOpt) Only replicate metadata, not images.
[DEFAULT] public_endpoint = None	(StrOpt) Public url to use for versions endpoint. The default is None, which will use the request's host_url attribute to populate the URL base. If Glance is operating behind a proxy, you will want to change this to represent the proxy's URL.
[DEFAULT] slavetoken =	(StrOpt) Pass in your authentication token if you have one. This is the token used for the slave.
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] token =	(StrOpt) Pass in your authentication token if you have one. If you use this option the same token is used for both the master and the slave.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
[database] backend = sqlalchemy	(StrOpt) The back end to use for the database.
[database] connection = None	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
[database] db_max_retries = 20	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
[database] db_max_retry_interval = 10	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
[database] db_retry_interval = 1	(IntOpt) Seconds between retries of a database transaction.
[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.

Option = default value	(Type) Help string
[database] max_retries = 10	(IntOpt) Maximum number of database connection retries during startup. Set to -1 to specify 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] slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
[database] sqlite_db = oslo.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.
[elasticsearch] hosts = 127.0.0.1:9200	(ListOpt) List of nodes where Elasticsearch instances are running. A single node should be defined as an IP address and port number.
[glance_store] store_capabilities_update_min_interval = 0	(IntOpt) Minimum interval seconds to execute updating dynamic storage capabilities based on backend status then. It's not a periodic routine, the update logic will be executed only when interval seconds elapsed and an operation of store has triggered. The feature will be enabled only when the option value greater then zero.
[glance_store] swift_store_cacert = None	(StrOpt) A string giving the CA certificate file to use in SSL connections for verifying certs.
[glance_store] swift_store_endpoint = None	(StrOpt) If set, the configured endpoint will be used. If None, the storage url from the auth response will be used.
[glance_store] swift_store_multiple_containers_seed = 0	(IntOpt) When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.Used only when swift_store_multi_tenant is disabled. The total number of containers that will be used is equal to 16^N, so if this config option is set to 2, then 16^2=256 containers will be used to store images.
[glance_store] vmware_datastores = None	(MultiStrOpt) A list of datastores where the image can be stored. This option may be specified multiple times for specifying multiple datastores. Either one of vmware_datastore_name or vmware_datastores is required. The datastore name should be specified after its datacenter path, seperated by ":". An optional weight may be given after the datastore name, seperated again by ":". Thus, the required format becomes <datacenter_path>:<datastore_name>:<optional_weight>. When adding an image, the datastore with highest weight will be selected, unless there is not enough free space available in cases where the image size is already known. If no weight is given, it is assumed to be zero and the directory will be considered for selection last. If multiple datastores have the same weight, then the one with the most free space available is selected.
[oslo_concurrency] disable_process_locking = False	(BoolOpt) Enables or disables inter-process locks.

Option = default value	(Type) Help string
[oslo_concurrency] lock_path = None	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_policy] policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
[oslo_policy] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[oslo_policy] policy_file = policy.json	(StrOpt) The JSON file that defines policies.
[service_credentials] insecure = False	(BoolOpt) Disables X.509 certificate validation when an SSL connection to Identity Service is established.
[service_credentials] os_auth_url = http://localhost:5000/v2.0	(StrOpt) Auth URL to use for OpenStack service access.
[service_credentials] os_cacert = None	(StrOpt) Certificate chain for SSL validation.
[service_credentials] os_endpoint_type = publicURL	(StrOpt) Type of endpoint in Identity service catalog to use for communication with OpenStack services.
[service_credentials] os_password = admin	(StrOpt) Password to use for OpenStack service access.

Option = default value	(Type) Help string
[service_credentials] os_region_name = None	(StrOpt) Region name to use for OpenStack service endpoints.
[service_credentials] os_tenant_id =	(StrOpt) Tenant ID to use for OpenStack service access.
[service_credentials] os_tenant_name = admin	(StrOpt) Tenant name to use for OpenStack service access.
[service_credentials] os_username = glance	(StrOpt) User name to use for OpenStack service access.
[task] work_dir = None	(StrOpt) Work dir for asynchronous task operations. The directory set here will be used to operate over images - normally before they are imported in the destination store. When providing work dir, make sure enough space is provided for concurrent tasks to run efficiently without running out of space. A rough estimation can be done by multiplying the number of `max_workers` - or the N of workers running - by an average image size (e.g 500MB). The image size estimation should be done based on the average size in your deployment. Note that depending on the tasks running you may need to multiply this number by some factor depending on what the task does. For example, you may want to double the available size if image conversion is enabled. All this being said, remember these are just estimations and you should do them based on the worst case scenario and be prepared to act in case they were wrong.
[taskflow_executor] conversion_format = None	(StrOpt) The format to which images will be automatically converted. Can be 'qcow2' or 'raw'.
[taskflow_executor] engine_mode = parallel	(StrOpt) The mode in which the engine will run. Can be 'serial' or 'parallel'.
[taskflow_executor] max_workers = 10	(IntOpt) The number of parallel activities executed at the same time by the engine. The value can be greater than one when the engine mode is 'parallel'.

Table 8.32. New default values

Option	Previous default value	New default value
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	local
[task] task_executor	eventlet	taskflow

Table 8.33. Deprecated options

Deprecated option	New Option
[DEFAULT] log-format	None
[glance_store] vmware_datacenter_path	None
[DEFAULT] log_format	None
[DEFAULT] use-syslog	None
[task] eventlet_executor_pool_size	[taskflow_executor] max_workers

Deprecated option	New Option
[glance_store] vmware_datastore_name	None
[DEFAULT] use_syslog	None

9. Networking

Table of Contents

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 9.1. Description of common configuration options

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.
`api_workers` = *0*	(IntOpt) Number of separate API 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
`core_plugin` = *None*	(StrOpt) The core plugin Neutron will use
`default_ipv4_subnet_pool` = *None*	(StrOpt) Default IPv4 subnet-pool to be used for automatic subnet CIDR allocation
`default_ipv6_subnet_pool` = *None*	(StrOpt) Default IPv6 subnet-pool to be used for automatic subnet CIDR allocation
`dhcp_agent_notification` = *True*	(BoolOpt) Allow sending resource operation notification to DHCP agent

Configuration option = Default value	Description
dhcp_agents_per_network = *1*	(IntOpt) Number of DHCP agents scheduled to host a network.
dhcp_broadcast_reply = *False*	(BoolOpt) Use broadcast in DHCP replies
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 (in seconds). Use -1 to tell dnsmasq to use infinite lease times.
dhcp_load_type = *networks*	(StrOpt) Representing the resource type whose load is being reported by the agent. This can be "networks", "subnets" or "ports". When specified (Default is networks), the server will extract particular load sent as part of its agent configuration object from the agent report state, which is the number of resources being consumed, at every report_interval.dhcp_load_type can be used in combination with network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.WeightScheduler When the network_scheduler_driver is WeightScheduler, dhcp_load_type can be configured to represent the choice for the resource being balanced. Example: dhcp_load_type=networks
enable_services_on_agents_with_admin_state_down = *False*	(BoolOpt) Enable services on an agent with admin_state_up False. If this option is False, when admin_state_up of an agent is turned False, services on it will be disabled. Agents with admin_state_up False are not selected for automatic scheduling regardless of this option. But manual scheduling to such agents is available if this option is True.
endpoint_type = *publicURL*	(StrOpt) Network service endpoint type to pull from the keystone catalog
force_gateway_on_subnet = *True*	(BoolOpt) Ensure that configured gateway is on subnet. For IPv6, validate only if gateway is not a link local address. Deprecated, to be removed during the K release, at which point the check will be mandatory.
host = *localhost*	(StrOpt) Hostname to be used by the neutron server, agents and services running on this machine. All the agents and services running on this machine must use the same host value.
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
mac_generation_retries = *16*	(IntOpt) How many times Neutron will retry MAC generation
max_allowed_address_pair = *10*	(IntOpt) Maximum number of allowed address pairs
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
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
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

Configuration option = Default value	Description
periodic_monitoring_interval = 5	(IntOpt) Periodic interval at which the plugin checks for the monitoring L2 gateway agent
report_interval = 300	(IntOpt) Interval between two metering reports
state_path = /var/lib/neutron	(StrOpt) Where to store Neutron state files. This directory must be writable by the agent.
vlan_transparent = False	(BoolOpt) If True, then allow plugins that support it to create VLAN transparent networks.
[AGENT]	
check_child_processes_action = respawn	(StrOpt) Action to be executed when a child process dies
check_child_processes_interval = 60	(IntOpt) Interval between checks of child process liveness (seconds), use 0 to disable
root_helper = sudo	(StrOpt) Root helper application.
root_helper_daemon = None	(StrOpt) Root helper daemon application to use when possible.
[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.
[keystone_authtoken]	
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.

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.

Note

The following plugins have been removed in Kilo:

- Ryu plugin. The Ryu team recommends that you migrate to the ML2 plugin with the ofagent mechanism driver. However, note that the functionality is not the same. There is no upgrade procedure currently available.
- Mellanox plugin.

BigSwitch configuration options

Table 9.2. Description of BigSwitch configuration options

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_distributed =	(ListOpt) Nova compute nodes to manually set VIF type to distributed
node_override_vif_dvs =	(ListOpt) Nova compute nodes to manually set VIF type to dvs
node_override_vif_hostdev =	(ListOpt) Nova compute nodes to manually set VIF type to hostdev
node_override_vif_hw_veb =	(ListOpt) Nova compute nodes to manually set VIF type to hw_veb
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
node_override_vif_vrouter =	(ListOpt) Nova compute nodes to manually set VIF type to vrouter
vif_type = *ivs*	(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, hw_veb, dvs, other, distributed, vrouter*	(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.

Configuration option = Default value	Description
consistency_interval = *60*	(IntOpt) Time between verifications that the backend controller database is consistent with Neutron. (0 to disable)
neutron_id = *neutron-images*	(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 oneserver is needed per deployment, but you may wish todeploy 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 = *ivs*	(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 9.3. Description of Brocade configuration options

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

Brocade MLX L3 plug-in

Configure switch names to be used as group names as described below

Table 9.4. Description of Brocade MLX L3 plug-in configuration options

Configuration option = Default value	Description
[l3_brocade_mlx]	
switch_names =	(StrOpt) Switches connected to the compute nodes

Brocade Vyatta layer 3 plug-in

The Brocade Vyatta Layer 3 plug-in configures Vyatta vRouter. More information about the plug-in is available at: Brocade_Vyatta_L3_Plugin.

Use the following options to configure the Brocade Vyatta Layer 3 plug-in.

Table 9.5. Description of Brocade Vyatta L3 plug-in configuration options

Configuration option = Default value	Description
[VROUTER]	
flavor = *2*	(StrOpt) Nova VM flavor for instances of Vyatta vRouter.
image_id = *None*	(StrOpt) Nova image id for instances of Vyatta vRouter.
keystone_url = *None*	(StrOpt) Keystone URL.
management_network_id = *None*	(StrOpt) Vyatta vRouter management network id.
nova_poll_interval = *5*	(IntOpt) Number of seconds between consecutive Nova queries when waiting for router instance status change.
nova_spawn_timeout = *300*	(IntOpt) Number of seconds to wait for Nova to activate instance before setting resource to error state.
tenant_admin_name = *None*	(StrOpt) Name of tenant admin user.
tenant_admin_password = *None*	(StrOpt) Tenant admin password.
tenant_id = *None*	(StrOpt) UUID of tenant that holds Vyatta vRouter instances.
vrouter_boot_timeout = *300*	(IntOpt) Number of seconds to wait for Vyatta vRouter to boot before setting resource to error state.
vrouter_credentials = *vyatta:vyatta*	(StrOpt) Vyatta vRouter login credentials
vrouter_poll_interval = *5*	(IntOpt) Number of seconds between consecutive Vyatta vRouter queries when waiting for router instance boot.

CISCO configuration options

Table 9.6. Description of Cisco configuration options

Configuration option = Default value	Description
[CISCO]	
model_class = *neutron.plugins.cisco.models.virt_phy_sw_v2.VirtualPhysicalSwitchModelV2*	(StrOpt) Model Class
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

Configuration option = Default value	Description
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
http_pool_size = *4*	(IntOpt) Number of threads to use to make HTTP requests
http_timeout = *15*	(IntOpt) N1K http timeout duration in seconds
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 = *60*	(IntOpt) N1K Policy profile polling duration in seconds
restrict_network_profiles = *True*	(BoolOpt) Restrict tenants from accessing network profiles belonging to some other tenant
restrict_policy_profiles = *False*	(BoolOpt) Restrict the visibility of policy profiles to the tenants
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
[cfg_agent]	
device_connection_timeout = *30*	(IntOpt) Time in seconds for connecting to a hosting device
hosting_device_dead_timeout = *300*	(IntOpt) The time in seconds until a backlogged hosting device is presumed dead. This value should be set up high enough to recover from a period of connectivity loss or high load when the device may not be responding.
[general]	
backlog_processing_interval = *10*	(IntOpt) Time in seconds between renewed scheduling attempts of non-scheduled routers.
cfg_agent_down_time = *60*	(IntOpt) Seconds of no status update until a cfg agent is considered down.
default_security_group = *mgmt_sec_grp*	(StrOpt) Default security group applied on management port. Default value is mgmt_sec_grp.
ensure_nova_running = *True*	(BoolOpt) Ensure that Nova is running before attempting to create any VM.
l3_admin_tenant = *L3AdminTenant*	(StrOpt) Name of the L3 admin tenant.
management_network = *osn_mgmt_nw*	(StrOpt) Name of management network for device configuration. Default value is osn_mgmt_nw
service_vm_config_path = */opt/stack/data/neutron/cisco/config_drive*	(StrOpt) Path to config drive files for service VM instances.
templates_path = */opt/stack/data/neutron/cisco/templates*	(StrOpt) Path to templates for hosting devices.
[hosting_devices]	
csr1kv_booting_time = *420*	(IntOpt) Booting time in seconds before a CSR1kv becomes operational.
csr1kv_cfgagent_router_driver = *networking_cisco.plugins.cisco.cfg_agent.device_drivers.csr1kv.csr1kv_routing_driver.CSR1kvRoutingDriv*	(StrOpt) Config agent driver for CSR1kv.

Configuration option = Default value	Description
csr1kv_device_driver = networking_cisco.plugins.cisco.l3.hosting_device_drivers.csr1kv_hd_driver.CSR1kvHostingDeviceDriver	(StrOpt) Hosting device driver for CSR1kv.
csr1kv_flavor = 621	(StrOpt) UUID of Nova flavor for CSR1kv.
csr1kv_image = csr1kv_openstack_img	(StrOpt) Name of Glance image for CSR1kv.
csr1kv_password = cisco	(StrOpt) Password to use for CSR1kv configurations.
csr1kv_plugging_driver = networking_cisco.plugins.cisco.l3.plugging_drivers.n1kv_trunking_driver.N1kvTrunkingPlugDriver	(StrOpt) Plugging driver for CSR1kv.
csr1kv_username = stack	(StrOpt) Username to use for CSR1kv configurations.

CloudBase Hyper-V Agent configuration options

Table 9.7. Description of HyperV agent configuration options

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 by 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 succeedes.
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]	
force_hyperv_utils_v1 = False	(BoolOpt) Force V1 WMI utility classes

Embrane configuration options

Table 9.8. Description of Embrane configuration options

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 9.9. Description of SDN-VE configuration options

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 = *OVERLAY*	(StrOpt) Tenant type: OVERLAY (default) or OF.
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> mappings.
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.
password = *admin*	(StrOpt) SDN-VE administrator password.
port = *8443*	(StrOpt) SDN-VE controller port number.
reset_bridge = *True*	(BoolOpt) Whether to reset the integration bridge before use.
use_fake_controller = *False*	(BoolOpt) Whether to use a fake controller.
userid = *admin*	(StrOpt) SDN-VE administrator user ID.
[SDNVE_AGENT]	
polling_interval = *2*	(IntOpt) Agent polling interval if necessary.
rpc = *True*	(BoolOpt) Whether to use rpc.

Layer 2 Gateway configuration options

Table 9.10. Description of Layer 2 Gateway configuration options

Configuration option = Default value	Description
[DEFAULT]	
default_device_name = *Switch1*	(StrOpt) default_device_name of the l2 gateway
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
l2gw_callback_class = *networking_l2gw.services.l2gateway.ovsdb.data.L2GatewayOVSDBCallbacks*	(StrOpt) L2 gateway plugin callback class where theRPCs from the agent are going to get invoked
quota_l2_gateway = *5*	(IntOpt) Number of l2 gateways allowed per tenant, -1 for unlimited
[ovsdb]	
l2_gw_agent_ca_cert_base_path = *None*	(StrOpt) Trusted issuer CA cert
l2_gw_agent_cert_base_path = *None*	(StrOpt) L2 gateway agent public certificate
l2_gw_agent_priv_key_base_path = *None*	(StrOpt) L2 gateway agent private key
max_connection_retries = *10*	(IntOpt) Maximum number of retries to open a socket with the OVSDB server

Configuration option = Default value	Description
ovsdb_hosts = *host1:127.0.0.1:6632*	(StrOpt) OVSDB server name:host/IP:port
periodic_interval = *20*	(IntOpt) Seconds between periodic task runs

Linux bridge Agent configuration options

Table 9.11. Description of Linux Bridge agent configuration options

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 = *True*	(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 = *None*	(IPOpt) 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.

Meta Plug-in configuration options

The Meta Plug-in allows you to use multiple plug-ins at the same time.

Table 9.12. Description of meta configuration options

Configuration option = Default value	Description
[META]	
default_flavor =	(StrOpt) Default flavor to use, when flavor:network is not specified at network creation.
default_l3_flavor =	(StrOpt) Default L3 flavor to use, when flavor:router is not specified at router creation. Ignored if 'l3_plugin_list' is blank.
extension_map =	(StrOpt) Comma separated list of method:flavor to select specific plugin for a method. This has priority over method search order based on 'plugin_list'.
l3_plugin_list =	(StrOpt) Comma separated list of flavor:neutron_plugin for L3 service plugins to load. This is intended for specifying L2 plugins which support L3 functions. If you use a router service plugin, set this blank.
plugin_list =	(StrOpt) Comma separated list of flavor:neutron_plugin for plugins to load. Extension method is searched in the list order and the first one is used.

Configuration option = Default value	Description
rpc_flavor =	(StrOpt) Specifies flavor for plugin to handle 'q-plugin' RPC requests.
supported_extension_aliases =	(StrOpt) Comma separated list of 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.

Configure MTU for VXLAN tunnelling

Specific MTU configuration is necessary for VXLAN to function as expected:

- One option 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 option 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 9.13. Description of ML2 configuration options

Configuration option = Default value	Description
[ml2]	
extension_drivers =	(ListOpt) An ordered list of extension driver entrypoints to be loaded from the neutron.ml2.extension_drivers namespace.
mechanism_drivers =	(ListOpt) An ordered list of networking mechanism driver entrypoints to be loaded from the neutron.ml2.mechanism_drivers namespace.
path_mtu = *0*	(IntOpt) The maximum permissible size of an unfragmented packet travelling from and to addresses where encapsulated Neutron traffic is sent. If <= 0, the path MTU is indeterminate.
physical_network_mtus =	(ListOpt) A list of mappings of physical networks to MTU values. The format of the mapping is <physnet>:<mtu val>. This mapping allows specifying a physical network MTU value that differs from the default segment_mtu value.
segment_mtu = *0*	(IntOpt) The maximum permissible size of an unfragmented packet travelling a L2 network segment. If <= 0, the segment MTU is indeterminate.
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 9.14. Description of ML2 Flat mechanism driver configuration options

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 9.15. Description of ML2 GRE configuration options

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 9.16. Description of ML2 VLAN configuration options

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 9.17. Description of ML2 VXLN configuration options

Configuration option = Default value	Description
[ml2_type_nexus_vxlan]	
`mcast_ranges` =	(ListOpt) List of multicast groups to be used for global VNIDsin the format - a:b,c,e:f.
`vni_ranges` =	(ListOpt) List of global VNID ranges in the format - a:b, c:d.Multiple ranges can be separated by a comma
[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 9.18. Description of ML2 Arista mechanism driver configuration options

Configuration option = Default value	Description
[ml2_arista]	

Configuration option = Default value	Description
eapi_host =	(StrOpt) Arista EOS IP address. This is required field. If not set, all communications to Arista EOS will fail.
eapi_password =	(StrOpt) Password for Arista EOS. This is required field. If not set, all communications to Arista EOS will fail.
eapi_username =	(StrOpt) Username for Arista EOS. This is required field. If not set, all communications to Arista EOS will fail.
region_name = *RegionOne*	(StrOpt) Defines Region Name that is assigned to this OpenStack Controller. This is useful when multiple OpenStack/Neutron controllers are managing the same Arista HW clusters. Note that this name must match with the region name registered (or known) to keystone service. Authentication with Keysotne is performed by EOS. This is optional. If not set, a value of "RegionOne" is assumed.
sync_interval = *180*	(IntOpt) Sync interval in seconds between Neutron plugin and EOS. This interval defines how often the synchronization is performed. This is an optional field. 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.

Table 9.19. Description of Arista layer-3 service plug-in configuration options

Configuration option = Default value	Description
[l3_arista]	
l3_sync_interval = *180*	(IntOpt) Sync interval in seconds between L3 Service plugin and EOS. This interval defines how often the synchronization is performed. This is an optional field. If not set, a value of 180 seconds is assumed
mlag_config = *False*	(BoolOpt) This flag is used indicate if Arista Switches are configured in MLAG mode. If yes, all L3 config is pushed to both the switches automatically. If this flag is set to True, ensure to specify IP addresses of both switches. This is optional. If not set, a value of "False" is assumed.
primary_l3_host =	(StrOpt) Arista EOS IP address. This is required field. If not set, all communications to Arista EOS will fail
primary_l3_host_password =	(StrOpt) Password for Arista EOS. This is required field. If not set, all communications to Arista EOS will fail
primary_l3_host_username =	(StrOpt) Username for Arista EOS. This is required field. If not set, all communications to Arista EOS will fail
secondary_l3_host =	(StrOpt) Arista EOS IP address for second Switch MLAGed with the first one. This an optional field, however, if mlag_config flag is set, then this is required. If not set, all communications to Arista EOS will fail
use_vrf = *False*	(BoolOpt) A "True" value for this flag indicates to create a router in VRF. If not set, all routers are created in default VRF. This is optional. If not set, a value of "False" is assumed.

Modular Layer 2 (ml2) BigSwitch Mechanism configuration options

Table 9.20. Description of ML2 BigSwitch mechanism driver configuration options

Configuration option = Default value	Description
[NOVA]	

Configuration option = Default value	Description
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_distributed =	(ListOpt) Nova compute nodes to manually set VIF type to distributed
node_override_vif_dvs =	(ListOpt) Nova compute nodes to manually set VIF type to dvs
node_override_vif_hostdev =	(ListOpt) Nova compute nodes to manually set VIF type to hostdev
node_override_vif_hw_veb =	(ListOpt) Nova compute nodes to manually set VIF type to hw_veb
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
node_override_vif_vrouter =	(ListOpt) Nova compute nodes to manually set VIF type to vrouter
vif_type = *ivs*	(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, hw_veb, dvs, other, distributed, vrouter*	(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. (0 to disable)
neutron_id = *neutron-images*	(StrOpt) User defined identifier for this Neutron deployment
no_ssl_validation = *False*	(BoolOpt) Disables SSL certificate validation for controllers

Configuration option = Default value	Description
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 oneserver is needed per deployment, but you may wish todeploy 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 = *ivs*	(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.

Modular Layer 2 (ml2) Brocade Mechanism configuration options

Table 9.21. Description of ML2 Brocade mechanism driver configuration options

Configuration option = Default value	Description
[ml2_brocade]	
address =	(StrOpt) The address of the host to SSH to
ostype = *NOS*	(StrOpt) OS Type of the switch
osversion = *4.0.0*	(StrOpt) OS Version number
password = *password*	(StrOpt) The SSH password to use
physical_networks =	(StrOpt) Allowed physical networks
rbridge_id = *1*	(StrOpt) Rbridge id of provider edge router(s)
username = *admin*	(StrOpt) The SSH username to use

Modular Layer 3 (ml2) Brocade MLX ICX Mechanism configuration options

Configure switch names to be used as group names as described below

Table 9.22. Description of ML2 Brocade MLX ICX mechanism driver configuration options

Configuration option = Default value	Description
[ml2_brocade_fi_ni]	
switch_names =	(StrOpt) Switches connected to the compute nodes

Modular Layer 2 (ml2) Cisco Mechanism configuration options

Table 9.23. Description of ML2 Cisco mechanism driver configuration options

Configuration option = Default value	Description
[DEFAULT]	
apic_system_id = *openstack*	(StrOpt) Prefix for APIC domain/names/profiles created
[ml2_cisco]	
host_key_checks = *False*	(BoolOpt) Enable strict host key checks when connecting to Nexus switches
managed_physical_network = *None*	(StrOpt) The physical network managed by the switches.
persistent_switch_config = *False*	(BoolOpt) To make Nexus configuration persistent
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
switch_heartbeat_time = *0*	(IntOpt) Periodic time to check switch connection. (0=disabled)
switch_replay_count = *3*	(IntOpt) Number of times to attempt config replay with switch.
vlan_name_prefix = *q-*	(StrOpt) VLAN Name prefix
vxlan_global_config = *False*	(BoolOpt) Create and delete Nexus switch VXLAN global settings; feature nv overlay, feature vn-segment-vlan-based, interface nve + source-interface loopback
[ml2_cisco_apic]	
apic_agent_poll_interval = *2*	(FloatOpt) Interval between agent poll for topology (in sec)
apic_agent_report_interval = *30*	(FloatOpt) Interval between agent status updates (in sec)
apic_app_profile_name = *${apic_system_id}_app*	(StrOpt) Name for the app profile used for Openstack
apic_domain_name = *${apic_system_id}*	(StrOpt) Name for the domain created on APIC
apic_entity_profile = *${apic_system_id}_entity_profile*	(StrOpt) Name of the entity profile to be created
apic_function_profile = *${apic_system_id}_function_profile*	(StrOpt) Name of the function profile to be created
apic_host_uplink_ports =	(ListOpt) The uplink ports to check for ACI connectivity
apic_hosts =	(ListOpt) An ordered list of host names or IP addresses of the APIC controller(s).
apic_lacp_profile = *${apic_system_id}_lacp_profile*	(StrOpt) Name of the LACP profile to be created
apic_name_mapping = *use_name*	(StrOpt) Name mapping strategy to use: use_uuid | use_name

Configuration option = Default value	Description
apic_node_profile = *${apic_system_id}_node_profile*	(StrOpt) Name of the node profile to be created
apic_password = *None*	(StrOpt) Password for the APIC controller
apic_sync_interval = *0*	(IntOpt) Synchronization interval in seconds
apic_use_ssl = *True*	(BoolOpt) Use SSL to connect to the APIC controller
apic_username = *None*	(StrOpt) Username for the APIC controller
apic_vlan_ns_name = *${apic_system_id}_vlan_ns*	(StrOpt) Name for the vlan namespace to be used for Openstack
apic_vlan_range = *2:4093*	(StrOpt) Range of VLAN's to be used for Openstack
apic_vpc_pairs =	(ListOpt) The switch pairs for VPC connectivity
[ml2_cisco_n1kv]	
default_policy_profile = *default-pp*	(StrOpt) Cisco Nexus1000V default policy profile.
default_vlan_network_profile = *default-vlan-np*	(StrOpt) Cisco Nexus1000V default network profile for VLAN networks.
default_vxlan_network_profile = *default-vxlan-np*	(StrOpt) Cisco Nexus1000V default network profile for VXLAN networks.
http_pool_size = *4*	(IntOpt) Number of threads to use to make HTTP requests.
http_timeout = *15*	(IntOpt) HTTP timeout, in seconds, for connections to the Cisco Nexus1000V VSMs.
n1kv_vsm_ips = *None*	(ListOpt) Comma Separated IP Addresses of the Cisco Nexus1000V VSMs.
password = *None*	(StrOpt) Password for all configured Cisco Nexus1000V VSMs.
poll_duration = *60*	(IntOpt) Cisco Nexus1000V policy profile polling duration in seconds.
restrict_policy_profiles = *False*	(BoolOpt) Restrict the visibility of policy profiles to the tenants.
sync_interval = *300*	(IntOpt) Time interval between consecutive neutron-VSM syncs.
username = *None*	(StrOpt) Username for all configured Cisco Nexus1000V VSMs.
[ml2_cisco_ucsm]	
supported_pci_devs = *1137:0071, 8086:10c9*	(ListOpt) List of comma separated vendor_id:product_id of SR_IOV capable devices supported by this MD. This MD supports both VM-FEX and SR-IOV devices.
ucsm_host_list = *None*	(ListOpt) List of comma separated Host:Service Profile tuples providing the Service Profile associated with each Host to be supported by this MD.
ucsm_ip = *None*	(StrOpt) Cisco UCS Manager IP address. This is a required field to communicate with a Cisco UCS Manager.
ucsm_password = *None*	(StrOpt) Password for UCS Manager. This is a required field to communicate with a Cisco UCS Manager.
ucsm_username = *None*	(StrOpt) Username for UCS Manager. This is a required field to communicate with a Cisco UCS Manager.

Modular Layer 2 (ml2) Freescale SDN Mechanism configuration options

Table 9.24. Description of ML2 Freescale SDN mechanism driver configuration options

Configuration option = Default value	Description
[ml2_fslsdn]	

Configuration option = Default value	Description
crd_api_insecure = *False*	(BoolOpt) If set, ignore any SSL validation issues.
crd_auth_strategy = *keystone*	(StrOpt) Auth strategy for connecting to neutron in admin context.
crd_auth_url = *http://127.0.0.1:5000/v2.0/*	(StrOpt) CRD Auth URL.
crd_ca_certificates_file = *None*	(StrOpt) Location of ca certificates file to use for CRD client requests.
crd_password = *password*	(StrOpt) CRD Service Password.
crd_region_name = *RegionOne*	(StrOpt) Region name for connecting to CRD Service in admin context.
crd_tenant_name = *service*	(StrOpt) CRD Tenant Name.
crd_url = *http://127.0.0.1:9797*	(StrOpt) URL for connecting to CRD service.
crd_url_timeout = *30*	(IntOpt) Timeout value for connecting to CRD service in seconds.
crd_user_name = *crd*	(StrOpt) CRD service Username.

Modular Layer 2 (ml2) OpenDaylight Mechanism configuration options

Table 9.25. Description of ML2 OpenDaylight mechanism driver configuration options

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 Agent (ofagent) Mechanism configuration options

Table 9.26. Description of ML2 ofagent mechanism driver configuration options

Configuration option = Default value	Description
[AGENT]	
dont_fragment = *True*	(BoolOpt) Set or un-set the don't fragment (DF) bit on outgoing IP packet carrying GRE/VXLAN tunnel.

Modular Layer 2 (ml2) L2 Population Mechanism configuration options

Table 9.27. Description of ML2 L2 population configuration options

Configuration option = Default value	Description
[l2pop]	
agent_boot_time = *180*	(IntOpt) Delay within which agent is expected to update existing ports whent it restarts

Modular Layer 2 (ml2) Tail-f NCS Mechanism configuration options

Table 9.28. Description of ML2 NCS mechanism driver configuration options

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

Modular Layer 2 (ml2) SR-IOV Mechanism configuration options

Table 9.29. Description of ML2 ML2 SR-IOV driver configuration options

Configuration option = Default value	Description
[ml2_sriov]	
agent_required = *False*	(BoolOpt) SRIOV neutron agent is required for port binding
supported_pci_vendor_devs = *15b3:1004, 8086:10ca*	(ListOpt) Supported PCI vendor devices, defined by vendor_id:product_id according to the PCI ID Repository. Default enables support for Intel and Mellanox SR-IOV capable NICs

MidoNet configuration options

Table 9.30. Description of Midonet configuration options

Configuration option = Default value	Description
[MIDONET]	
cluster_ip = *localhost*	(StrOpt) IP that the cluster service can be reached on
cluster_port = *8088*	(StrOpt) Port that the cluster service can be reached on
tunnel_protocol = *vxlan*	(StrOpt) Tunnel protocol used by Midonet
use_cluster = *False*	(BoolOpt) Use the midonet plugin that interacts with the tasks-based architecture, rather than the plugin that interacts with the HTTP based architecture.

NEC configuration options

Table 9.31. Description of Nec configuration options

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.
cert_file = *None*	(StrOpt) Location of certificate file.
driver = *trema*	(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.

Configuration option = Default value	Description
key_file = *None*	(StrOpt) Location of 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.
support_packet_filter_on_ofc_router = *True*	(BoolOpt) Support packet filter on OFC router interface.
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

One Convergence NVSD configuration options

Table 9.32. Description of NVSD driver configuration options

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
request_timeout = *30*	(IntOpt) NVSD controller REST API request timeout in seconds

OpenContrail configuration options

Table 9.33. Description of OpenContrail configuration options

Configuration option = Default value	Description
[CONTRAIL]	
api_server_ip = *127.0.0.1*	(StrOpt) IP address to connect to opencontrail controller
api_server_port = *8082*	(IntOpt) Port to connect to opencontrail controller

Open vSwitch Agent configuration options

Table 9.34. Description of Open vSwitch agent configuration options

Configuration option = Default value	Description
[DEFAULT]	
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]	

Configuration option = Default value	Description
arp_responder = *False*	(BoolOpt) Enable local ARP responder if it is supported. Requires OVS 2.1 and ML2 l2population driver. Allows the switch (when supporting an overlay) to respond to an ARP request locally without performing a costly ARP broadcast into the overlay.
dont_fragment = *True*	(BoolOpt) Set or un-set the don't fragment (DF) bit on outgoing IP packet carrying GRE/VXLAN tunnel.
enable_distributed_routing = *False*	(BoolOpt) Make the l2 agent run in DVR mode.
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 ovsdb for interface changes.
ovsdb_monitor_respawn_interval = *30*	(IntOpt) The number of seconds to wait before respawning the ovsdb monitor after losing communication with it.
prevent_arp_spoofing = *False*	(BoolOpt) Enable suppression of ARP responses that don't match an IP address that belongs to the port from which they originate. Note: This prevents the VMs attached to this agent from spoofing, it doesn't protect them from other devices which have the capability to spoof (e.g. bare metal or VMs attached to agents without this flag set to True). Spoofing rules will not be added to any ports that have port security disabled. This requires a version of OVS that supports matching ARP headers.
quitting_rpc_timeout = *10*	(IntOpt) Set new timeout in seconds for new rpc calls after agent receives SIGTERM. If value is set to 0, rpc timeout won't be changed
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>. Deprecated for ofagent.
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 = *None*	(IPOpt) Local IP address of tunnel endpoint.
ovsdb_interface = *vsctl*	(StrOpt) The interface for interacting with the OVSDB
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.
use_veth_interconnection = *False*	(BoolOpt) Use veths instead of patch ports to interconnect the integration bridge to physical bridges.

PLUMgrid configuration options

Table 9.35. Description of PLUMgrid configuration options

Configuration option = Default value	Description
[plumgriddirector]	
director_server = *localhost*	(StrOpt) PLUMgrid Director server to connect to

Configuration option = Default value	Description
director_server_port = *8080*	(StrOpt) PLUMgrid Director server port to connect to
driver = *networking_plumgrid.neutron.plugins.drivers.plumlib.Plumlib*	(StrOpt) PLUMgrid Driver
password = *password*	(StrOpt) PLUMgrid Director admin password
servertimeout = *5*	(IntOpt) PLUMgrid Director server timeout
username = *username*	(StrOpt) PLUMgrid Director admin username

SR-IOV configuration options

Table 9.36. Description of SR-IOV configuration options

Configuration option = Default value	Description
[SRIOV_NIC]	
exclude_devices =	(ListOpt) List of <network_device>:<excluded_devices> mapping network_device to the agent's node-specific list of virtual functions that should not be used for virtual networking. excluded_devices is a semicolon separated list of virtual functions (BDF format).to exclude from network_device. The network_device in the mapping should appear in the physical_device_mappings list.
physical_device_mappings =	(ListOpt) List of <physical_network>:<network_device> mapping physical network names to the agent's node-specific physical network device of SR-IOV physical function to be used for VLAN networks. All physical networks listed in network_vlan_ranges on the server should have mappings to appropriate interfaces on each agent

VMware vSphere configuration options

Table 9.37. Description of VMware configuration options

Configuration option = Default value	Description
[DEFAULT]	
default_interface_name = *breth0*	(StrOpt) Name of the interface on a L2 Gateway transport nodewhich should be used by default when setting up a network connection
[OVSVAPP]	
agent_driver = *networking_vsphere.agent.ovsvapp_agent.OVSvAppL2Agent*	(StrOpt) OVSvApp Agent implementation.
agent_maintenance = *False*	(BoolOpt) Turn on this flag during agent updates to help prevent datapath outage.
bridge_mappings =	(ListOpt) Bridge mappings.
dont_fragment = *True*	(IntOpt) Do not fragment.
integration_bridge = *br-int*	(StrOpt) Integration Bridge.
local_ip =	(StrOpt) Local IP address of VXLAN tunnel endpoint.
network_manager = *networking_vsphere.drivers.manager.VcenterManager*	(StrOpt) Driver Manager implementation for NetworkDriver
polling_interval = *2*	(IntOpt) The number of seconds the agent will wait between polling for local device changes.
report_interval = *4*	(IntOpt) Seconds between nodes reporting state to server.
tenant_network_type = *vlan*	(StrOpt) Network type for tenant networks
tunnel_bridge = *br-tun*	(StrOpt) Tunnel Bridge for tunneling.

Configuration option = Default value	Description
tunnel_types = *vxlan*	(ListOpt) Tunnel network types supported by the OVS-vApp Agent.
veth_mtu = *1500*	(IntOpt) MTU size of veth interfaces.
vxlan_udp_port = *4789*	(IntOpt) The UDP port to use for VXLAN tunnels.
[VMWARE]	
cert_check = *False*	(BoolOpt) Enable SSL certificate check for vCenter.
cert_path = */etc/ssl/certs/certs.pem*	(StrOpt) Certificate chain path containing cacert of vCenters.
cluster_dvs_mapping = *['']*	(MultiStrOpt) vCenter cluster to DVS mapping.
esx_hostname = *None*	(StrOpt) ESX host name where this OVSvApp is hosted.
esx_maintenance_mode = *True*	(BoolOpt) Set host into maintenance mode.
https_port = *443*	(IntOpt) Customized https_port for vCenter communication.
vcenter_api_retry_count = *5*	(StrOpt) Number of retries while connecting to vcenter server.
vcenter_id = *None*	(StrOpt) Unique ID of the vCenter Server on which this OVSvApp ishosted
vcenter_ip = *None*	(StrOpt) vCenter server IP.
vcenter_password = *None*	(StrOpt) vCenter server password.
vcenter_username = *None*	(StrOpt) vCenter server user name.
wsdl_location = *None*	(StrOpt) vCenter server wsdl location.

VMware NSX configuration options

Table 9.38. Description of VMware NSX configuration options

Configuration option = Default value	Description
[DEFAULT]	
conn_idle_timeout = *900*	(IntOpt) Reconnect connection to nsx if not used within this amount of time.
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 = *75*	(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
retries = *2*	(IntOpt) Number of time a request should be retried
[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 tranport type to use (stt, gre, bridge, ipsec_gre, or ipsec_stt)

Configuration option = Default value	Description
max_lp_per_bridged_ls = *5000*	(IntOpt) Maximum number of ports of a logical switch on a bridged transport zone (default 5000)
max_lp_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)
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). It must be set to 'service' for leveraging distributed routers.
[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
[nsx_v3]	
default_edge_cluster_uuid = *None*	(StrOpt) Default edge cluster identifier
[nsxv]	
backup_edge_pool = *service:large:4:10, service:compact:4:10, vdr:large:4:10*	(ListOpt) Defines edge pool using the format: <edge_type>:[edge_size]:<min_edges>:<max_edges>.edge_type: service,vdr. edge_size: compact, large, xlarge, quadlarge and default is large.
cluster_moid =	(ListOpt) Parameter listing the IDs of the clusters which are used by OpenStack.
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
dvs_id = *None*	(StrOpt) DVS ID for VLANs
external_network = *None*	(StrOpt) Network ID for physical network connectivity
manager_uri = *None*	(StrOpt) uri for vsm
maximum_tunnels_per_vnic = *20*	(IntOpt) Maximum number of sub interfaces supported per vnic in edge. The value should be in 1-110.
metadata_shared_secret = *None*	(StrOpt) Shared secret to sign metadata requests
mgt_net_default_gateway = *None*	(StrOpt) Management network default gateway for metadata proxy

Configuration option = Default value	Description
mgt_net_moid = *None*	(StrOpt) Network ID for management network connectivity
mgt_net_proxy_ips = *None*	(ListOpt) Management network IP address for metadata proxy
mgt_net_proxy_netmask = *None*	(StrOpt) Management network netmask for metadata proxy
nova_metadata_ips = *None*	(ListOpt) IP addresses used by Nova metadata service
nova_metadata_port = *8775*	(IntOpt) TCP Port used by Nova metadata server
password = *default*	(StrOpt) Password for vsm
resource_pool_id = *None*	(StrOpt) Optional parameter identifying the ID of resource to deploy NSX Edges
retries = *10*	(IntOpt) Maximum number of API retries on endpoint.
spoofguard_enabled = *True*	(BoolOpt) If True then plugin will use NSXV spoofguard component for port-security feature.
task_status_check_interval = *2000*	(IntOpt) Task status check interval
tenant_router_types = *shared, distributed, exclusive*	(ListOpt) Ordered list of router_types to allocate as tenant routers.
user = *admin*	(StrOpt) User name for vsm
vdn_scope_id = *None*	(StrOpt) Network scope ID for VXLAN virtual wires

Load-Balancer-as-a-Service Agent configuration options

Use the following options in the `lbaas_agent.ini` file for the LBaaS agent.

Table 9.39. Description of Load-Balancer-as-a-Service agent configuration options

Configuration option = Default value	Description
[DEFAULT]	
periodic_interval = *10*	(IntOpt) Seconds between periodic task runs.
Variable = *neutron_lbaas.services.loadbalancer.drivers.haproxy.namespace_driver.HaproxyNSDriver*	(MultiStrOpt) Drivers used to manage loadbalancing devices.
[haproxy]	
loadbalancer_state_path = *$state_path/lbaas*	(StrOpt) Location to store config and state files.
user_group = *nogroup*	(StrOpt) The user group.
Variable = *3*	(IntOpt) When delete and re-add the same vip, send this many gratuitous ARPs to flush the ARP cache in the Router. Set it below or equal to 0 to disable this feature.

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.openstack.common.notifier.rpc_notifier` in the `neutron.conf` file:

Table 9.40. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
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_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.

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`* parameter 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 9.41. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	

Configuration option = Default value	Description
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
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_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

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 9.42. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 9.43. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
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
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout

Table 9.44. Description of Redis configuration options

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 9.45. Description of AMQP configuration options

Configuration option = Default value	Description
[DEFAULT]	
control_exchange = *neutron*	(StrOpt) The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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.

Agent

Use the following options to alter agent-related settings.

Table 9.46. Description of agent configuration options

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 9.47. Description of API configuration options

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
backlog = *4096*	(IntOpt) Number of backlog requests to configure the socket with
client_socket_timeout = *900*	(IntOpt) Timeout for client connections socket operations. If an incoming connection is idle for this number of seconds it will be closed. A value of '0' means wait forever.
max_header_line = *16384*	(IntOpt) Max header line to accommodate large tokens
pagination_max_limit = *-1*	(StrOpt) The maximum number of items returned in a single response, value was 'infinite' or negative integer means no limit
retry_until_window = *30*	(IntOpt) Number of seconds to keep retrying to listen
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
tcp_keepidle = *600*	(IntOpt) Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.

Configuration option = Default value	Description
wsgi_keep_alive = *True*	(BoolOpt) 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.
[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 9.48. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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 un-

Configuration option = Default value	Description
	known 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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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.

Configuration option = Default value	Description
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.

Compute

Use the following options to alter Compute-related settings.

Table 9.49. Description of Compute configuration options

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. Deprecated in favour of an auth plugin in [nova].
nova_admin_password = *None*	(StrOpt) Password for connection to nova in admin context. Deprecated in favour of an auth plugin in [nova].
nova_admin_tenant_id = *None*	(StrOpt) The uuid of the admin nova tenant. Deprecated in favour of an auth plugin in [nova].
nova_admin_tenant_name = *None*	(StrOpt) The name of the admin nova tenant. Deprecated in favour of an auth plugin in [nova].
nova_admin_username = *None*	(StrOpt) Username for connecting to nova in admin context. Deprecated in favour of an auth plugin in [nova].
nova_client_cert =	(StrOpt) Client certificate for nova metadata api server.
nova_client_priv_key =	(StrOpt) Private key of client certificate.
nova_url = *http://127.0.0.1:8774/v2*	(StrOpt) URL for connection to nova. Deprecated in favour of an auth plugin in [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 9.50. Description of database configuration options

Configuration option = Default value	Description
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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.

Configuration option = Default value	Description
db_inc_retry_interval = *True*	(BoolOpt) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify 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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.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.

Logging

Use the following options to alter debug settings.

Table 9.51. Description of logging configuration options

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.

DHCP agent

Use the following options to alter Database-related settings.

Table 9.52. Description of DHCP agent configuration options

Configuration option = Default value	Description
[DEFAULT]	
advertise_mtu = *False*	(BoolOpt) If True, effort is made to advertise MTU settings to VMs via network methods (DHCP and RA MTU options) when the network's preferred MTU is known.
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. This option is deprecated and will be removed in a future release.

Distributed virtual router

Use the following options to alter DVR-related settings.

Table 9.53. Description of DVR configuration options

Configuration option = Default value	Description
[DEFAULT]	
dvr_base_mac = *fa:16:3f:00:00:00*	(StrOpt) The base mac address used for unique DVR instances by Neutron
router_distributed = *False*	(BoolOpt) System-wide flag to determine the type of router that tenants can create. Only admin can override.

Firewall-as-a-Service driver

Use the following options in the `fwaas_driver.ini` file for the FWaaS driver.

Table 9.54. Description of FwaaS configuration options

Configuration option = Default value	Description
[fwaas]	
enabled = *False*	(BoolOpt) Enable FWaaS

IPv6 router advertisement

Use the following options to alter IPv6 RA settings.

Table 9.55. Description of IPv6 router advertisement configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
ra_confs = *$state_path/ra*	(StrOpt) Location to store IPv6 RA config files

L3 agent

Use the following options in the `l3_agent.ini` file for the L3 agent.

Table 9.56. Description of L3 agent configuration options

Configuration option = Default value	Description
[DEFAULT]	
agent_mode = *legacy*	(StrOpt) The working mode for the agent. Allowed modes are: 'legacy' - this preserves the existing behavior where the L3 agent is deployed on a centralized networking node to provide L3 services like DNAT, and SNAT. Use this mode if you do not want to adopt DVR. 'dvr' - this mode enables DVR functionality and must be used for an L3 agent that runs on a compute host. 'dvr_snat' - this enables centralized SNAT support in conjunction with DVR. This mode must be used for an L3 agent running on a centralized node (or in single-host deployments, e.g. devstack)
allow_automatic_dhcp_failover = *True*	(BoolOpt) Automatically remove networks from offline DHCP agents.
allow_automatic_l3agent_failover = *False*	(BoolOpt) Automatically reschedule routers from offline L3 agents to online L3 agents.
enable_metadata_proxy = *True*	(BoolOpt) Allow running metadata proxy.
external_ingress_mark = *0x2*	(StrOpt) Iptables mangle mark used to mark ingress from external network
external_network_bridge = *br-ex*	(StrOpt) Name of bridge used for external network traffic.
gateway_external_network_id =	(StrOpt) UUID of external network for routers implemented by the agents.
ha_confs_path = *$state_path/ha_confs*	(StrOpt) Location to store keepalived/conntrackd config files
ha_vrrp_advert_int = *2*	(IntOpt) The advertisement interval in seconds
ha_vrrp_auth_password = *None*	(StrOpt) VRRP authentication password
ha_vrrp_auth_type = *PASS*	(StrOpt) VRRP authentication type
handle_internal_only_routers = *True*	(BoolOpt) Agent should implement routers with no gateway
ipv6_gateway =	(StrOpt) With IPv6, the network used for the external gateway does not need to have an associated subnet, since the automatically assigned link-local address (LLA) can be used. However, an IPv6 gateway address is needed for use as the next-hop for the default route. If no IPv6 gateway address is configured here, (and only then) the neutron router will be configured to get its default route from router advertisements (RAs) from the upstream router; in which case the upstream router must also be configured to send these RAs. The ipv6_gateway, when configured, should be the LLA of the interface on the upstream router. If a next-hop using a global unique address (GUA) is desired, it needs to be done via a subnet allocated to the network and not through this parameter.
l3_ha = *False*	(BoolOpt) Enable HA mode for virtual routers.
l3_ha_net_cidr = *169.254.192.0/18*	(StrOpt) Subnet used for the l3 HA admin network.

Configuration option = Default value	Description
max_l3_agents_per_router = *3*	(IntOpt) Maximum number of agents on which a router will be scheduled.
min_l3_agents_per_router = *2*	(IntOpt) Minimum number of agents on which a router will be scheduled.
router_id =	(StrOpt) If namespaces is disabled, the l3 agent can only configure a router that has the matching router ID.
send_arp_for_ha = *3*	(IntOpt) Send this many gratuitous ARPs for HA setup, if less than or equal to 0, the feature is disabled
[AGENT]	
comment_iptables_rules = *True*	(BoolOpt) Add comments to iptables rules.
use_helper_for_ns_read = *True*	(BoolOpt) Use the root helper to read the namespaces from the operating system.

Logging

Use the following options to alter logging settings.

Table 9.57. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.

Configuration option = Default value	Description
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
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. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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).

Metadata Agent

Use the following options in the `metadata_agent.ini` file for the Metadata agent.

Table 9.58. Description of metadata configuration options

Configuration option = Default value	Description
[DEFAULT]	
meta_flavor_driver_mappings = *None*	(StrOpt) Mapping between flavor and LinuxInterfaceDriver. It is specific to MetaInterfaceDriver used with admin_user, admin_password, admin_tenant_name, admin_url, auth_strategy, auth_region and endpoint_type.
metadata_access_mark = *0x1*	(StrOpt) Iptables mangle mark used to mark metadata valid requests
metadata_backlog = *4096*	(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_group =	(StrOpt) Group (gid or name) running metadata proxy after its initialization (if empty: agent effective group).
metadata_proxy_shared_secret =	(StrOpt) Shared secret to sign instance-id request
metadata_proxy_socket = *$state_path/metadata_proxy*	(StrOpt) Location for Metadata Proxy UNIX domain socket.
metadata_proxy_socket_mode = *deduce*	(StrOpt) Metadata Proxy UNIX domain socket mode, 3 values allowed: 'deduce': deduce mode from metadata_proxy_user/group values, 'user': set metadata proxy socket mode to 0o644, to use when metadata_proxy_user is agent effective user or root, 'group': set metadata proxy socket mode to 0o664, to use when metadata_proxy_group is agent effective group or root, 'all': set metadata proxy socket mode to 0o666, to use otherwise.
metadata_proxy_user =	(StrOpt) User (uid or name) running metadata proxy after its initialization (if empty: agent effective user).
metadata_proxy_watch_log = *None*	(BoolOpt) Enable/Disable log watch by metadata proxy. It should be disabled when metadata_proxy_user/group is not allowed to read/write its log file and copytruncate logrotate option must be used if logrotate is enabled on metadata proxy log files. Option default value is deduced from metadata_proxy_user: watch log is enabled if metadata_proxy_user is agent effective user id/name.
metadata_workers = *2*	(IntOpt) Number of separate worker processes for metadata server
nova_metadata_insecure = *False*	(BoolOpt) Allow to perform insecure SSL (https) requests to nova metadata
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.
nova_metadata_protocol = *http*	(StrOpt) Protocol to access nova metadata, http or https

Note

Previously, neutron metadata agent connected to a neutron server via REST API using a neutron client. This is ineffective because keystone is then fully involved into the authentication process and gets overloaded.

The neutron metadata agent has been reworked to use RPC by default to connect to a server since Kilo release. This is a typical way of interacting between neutron server and its agents. If neutron server does not support metadata RPC then neutron client will be used.

Warning

Do not run the `neutron-ns-metadata-proxy` proxy namespace as root on a node with the L3 agent running. In OpenStack Kilo and newer, you can change the permissions of `neutron-ns-metadata-proxy` after the proxy installation using the `metadata_proxy_user` and `metadata_proxy_group` options.

Metering Agent

Use the following options in the `metering_agent.ini` file for the Metering agent.

Table 9.59. Description of metering agent configuration options

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]	
`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.

Nova

Use the following options in the `neutron.conf` file to change nova-related settings.

Table 9.60. Description of nova configuration options

Configuration option = Default value	Description
[nova]	
`auth_plugin` = *`None`*	(StrOpt) Name of the plugin to load
`auth_section` = *`None`*	(StrOpt) Config Section from which to load plugin specific options
`cafile` = *`None`*	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
`certfile` = *`None`*	(StrOpt) PEM encoded client certificate cert file
`insecure` = *`False`*	(BoolOpt) Verify HTTPS connections.
`keyfile` = *`None`*	(StrOpt) PEM encoded client certificate key file
`region_name` = *`None`*	(StrOpt) Name of nova region to use. Useful if keystone manages more than one region.
`timeout` = *`None`*	(IntOpt) Timeout value for http requests

Policy

Use the following options in the `neutron.conf` file to change oslo middleware settings.

Table 9.61. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	

Configuration option = Default value	Description
max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.

Policy

Use the following options in the neutron.conf file to change policy settings.

Table 9.62. Description of policy configuration options

Configuration option = Default value	Description
[DEFAULT]	
allow_overlapping_ips = False	(BoolOpt) Allow overlapping IP support in Neutron
policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.json	(StrOpt) The JSON file that defines policies.

Quotas

Use the following options in the neutron.conf file for the quota system.

Table 9.63. Description of quotas configuration options

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_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.

Configuration option = Default value	Description
quota_port = *50*	(IntOpt) Number of ports allowed per tenant. A negative value means unlimited.
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.

Scheduler

Use the following options in the `neutron.conf` file to change scheduler settings.

Table 9.64. Description of scheduler configuration options

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 9.65. Description of security groups configuration options

Configuration option = Default value	Description
[SECURITYGROUP]	
defer_apply = *True*	(BoolOpt) Enable defer_apply on security bridge.
enable_ipset = *True*	(BoolOpt) Use ipset to speed-up the iptables based security groups.
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
ovsvapp_firewall_driver = *networking_vsphere.drivers.ovs_firewall.OVSFirewallDriver*	(StrOpt) DriverManager implementation for OVS based Firewall
security_bridge_mapping = *br-sec*	(StrOpt) <security_bridge>:<phy_interface>

Note

Now Networking uses iptables to achieve security group functions. In L2 agent with `enable_ipset` option enabled, it makes use of IPset to improve security group's performance, as it represents a hash set which is insensitive to the number of elements.

When a port is created, L2 agent will add an additional IPset chain to it's iptables chain, if the security group that this port belongs to has rules between other security group, the member of that security group will be added to the ipset chain.

If a member of a security group is changed, it used to reload iptables rules which is expensive. However, when IPset option is enabled on L2 agent, it does not need to reload iptables if only members of security group were changed, it should just update an IPset.

Note

A single default security group has been introduced in order to avoid race conditions when creating a tenant's default security group. The race conditions are caused by the uniqueness check of a new security group name. A table `default_security_group` implements such a group. It has `tenant_id` field as a primary key and `security_group_id`, which is an identifier of a default security group. The migration that introduces this table has a sanity check that verifies if a default security group is not duplicated in any tenant.

SSL and Certification Authority

Use the following options in the `neutron.conf` file to enable SSL.

Table 9.66. Description of CA and SSL configuration options

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

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 9.67. Log files used by Networking services

Log file	Service/interface
`dhcp-agent.log`	`neutron-dhcp-agent`
`l3-agent.log`	`neutron-l3-agent`

Log file	Service/interface
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

[a]The 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

# =========Start Global Config Option for Distributed L3 Router===============
# Setting the "router_distributed" flag to "True" will default to the creation
# of distributed tenant routers. The admin can override this flag by
 specifying
# the type of the router on the create request (admin-only attribute). Default
# value is "False" to support legacy mode (centralized) routers.
#
# router_distributed = False
#
# ===========End Global Config Option for Distributed L3 Router===============

# 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

# 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 the 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

# (StrOpt) Hostname to be used by the neutron server, agents and services
# running on this machine. All the agents and services running on this machine
# must use the same host value.
# The default value is hostname of the machine.
#
# host =

# 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
# 4h 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

# DVR 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. The 'dvr_base_mac' *must* be different from 'base_mac' to
# avoid mixing them up with MAC's allocated for tenant ports.
# A 4 octet example would be dvr_base_mac = fa:16:3f:4f:00:00
# The default is 3 octet
# dvr_base_mac = fa:16:3f:00:00:00

# Maximum amount of retries to generate a unique MAC address
# mac_generation_retries = 16

# DHCP Lease duration (in seconds).  Use -1 to
# tell dnsmasq to use infinite lease times.
# 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. For IPv6, validate only if
# gateway is not a link local address. Deprecated, to be removed during the
# K release, at which point the check will be mandatory.
# force_gateway_on_subnet = True

# Default maximum number of items returned in a single response,
# value == infinite and value < 0 means no max limit, and value must
# be greater than 0. If the number of items requested is greater than
# pagination_max_limit, server will just return pagination_max_limit
# of number of items.
# pagination_max_limit = -1

# Maximum number of DNS nameservers per subnet
# max_dns_nameservers = 5

# Maximum number of host routes per subnet
# max_subnet_host_routes = 20

# Maximum number of fixed ips per port
# max_fixed_ips_per_port = 5

# Maximum number of routes per router
# max_routes = 30

# Default Subnet Pool to be used for IPv4 subnet-allocation.
# Specifies by UUID the pool to be used in case of subnet-create being called
# without a subnet-pool ID.  The default of None means that no pool will be
# used unless passed explicitly to subnet create.  If no pool is used, then a
# CIDR must be passed to create a subnet and that subnet will not be allocated
# from any pool; it will be considered part of the tenant's private address
```

```
# space.
# default_ipv4_subnet_pool =

# Default Subnet Pool to be used for IPv6 subnet-allocation.
# Specifies by UUID the pool to be used in case of subnet-create being
# called without a subnet-pool ID.  Set to "prefix_delegation"
# to enable IPv6 Prefix Delegation in a PD-capable environment.
# See the description for default_ipv4_subnet_pool for more information.
# default_ipv6_subnet_pool =

# ========== items for MTU selection and advertisement =============
# Advertise MTU.  If True, effort is made to advertise MTU
# settings to VMs via network methods (ie. DHCP and RA MTU options)
# when the network's preferred MTU is known.
# advertise_mtu = False
# ======== end of items for MTU selection and advertisement =========

# ========== items for agent management extension =============
# Seconds to regard the agent as down; should be at least twice
# report_interval, to be sure the agent is down for good
# agent_down_time = 75
# ==========  end of items for agent management extension =====

# ========== items for agent scheduler extension =============
# Driver to use for scheduling network to DHCP agent
# network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.
ChanceScheduler
# Driver to use for scheduling router to a default L3 agent
# router_scheduler_driver = neutron.scheduler.l3_agent_scheduler.
ChanceScheduler
# Driver to use for scheduling a loadbalancer pool to an lbaas agent
# loadbalancer_pool_scheduler_driver = neutron.services.loadbalancer.
agent_scheduler.ChanceScheduler

# (StrOpt) Representing the resource type whose load is being reported by
# the agent.
# This can be 'networks','subnets' or 'ports'. When specified (Default is
 networks),
# the server will extract particular load sent as part of its agent
 configuration object
# from the agent report state, which is the number of resources being
 consumed, at
# every report_interval.
# dhcp_load_type can be used in combination with network_scheduler_driver =
# neutron.scheduler.dhcp_agent_scheduler.WeightScheduler
# When the network_scheduler_driver is WeightScheduler, dhcp_load_type can
# be configured to represent the choice for the resource being balanced.
# Example: dhcp_load_type = networks
# Values:
#   networks - number of networks hosted on the agent
#   subnets -  number of subnets associated with the networks hosted on the
 agent
#   ports   -  number of ports associated with the networks hosted on the
 agent
# dhcp_load_type = networks

# Allow auto scheduling networks to DHCP agent. It will schedule non-hosted
# networks to first DHCP agent which sends get_active_networks message to
# neutron server
# network_auto_schedule = True
```

```
# Allow auto scheduling routers to L3 agent. It will schedule non-hosted
# routers to first L3 agent which sends sync_routers message to neutron server
# router_auto_schedule = True

# Allow automatic rescheduling of routers from dead L3 agents with
# admin_state_up set to True to alive agents.
# allow_automatic_l3agent_failover = False

# Allow automatic removal of networks from dead DHCP agents with
# admin_state_up set to True.
# Networks could then be rescheduled if network_auto_schedule is True
# allow_automatic_dhcp_failover = True

# Number of DHCP agents scheduled to host a network. This enables redundant
# DHCP agents for configured networks.
# dhcp_agents_per_network = 1

# Enable services on agents with admin_state_up False.
# If this option is False, when admin_state_up of an agent is turned to
# False, services on it will be disabled. If this option is True, services
# on agents with admin_state_up False keep available and manual scheduling
# to such agents is available. Agents with admin_state_up False are not
# selected for automatic scheduling regardless of this option.
# enable_services_on_agents_with_admin_state_down = False

# ===========  end of items for agent scheduler extension =====

# ========== items for l3 extension ==============
# Enable high availability for virtual routers.
# l3_ha = False
#
# Maximum number of l3 agents which a HA router will be scheduled on. If it
# is set to 0 the router will be scheduled on every agent.
# max_l3_agents_per_router = 3
#
# Minimum number of l3 agents which a HA router will be scheduled on. The
# default value is 2.
# min_l3_agents_per_router = 2
#
# CIDR of the administrative network if HA mode is enabled
# l3_ha_net_cidr = 169.254.192.0/18
# ========== end of items for l3 extension =======

# ========== items for metadata proxy configuration ==============
# User (uid or name) running metadata proxy after its initialization
# (if empty: agent effective user)
# metadata_proxy_user =

# Group (gid or name) running metadata proxy after its initialization
# (if empty: agent effective group)
# metadata_proxy_group =

# Enable/Disable log watch by metadata proxy, it should be disabled when
# metadata_proxy_user/group is not allowed to read/write its log file and
# 'copytruncate' logrotate option must be used if logrotate is enabled on
# metadata proxy log files. Option default value is deduced from
# metadata_proxy_user: watch log is enabled if metadata_proxy_user is agent
# effective user id/name.
# metadata_proxy_watch_log =
```

```
# Location of Metadata Proxy UNIX domain socket
# metadata_proxy_socket = $state_path/metadata_proxy
# =========== end of items for metadata proxy configuration ==============

# ========== items for VLAN trunking networks ==========
# Setting this flag to True will allow plugins that support it to
# create VLAN transparent networks. This flag has no effect for
# plugins that do not support VLAN transparent networks.
# vlan_transparent = False
# ========== end of items for VLAN trunking networks ==========

# =========== WSGI parameters related to the API server ==============
# Number of separate worker processes to spawn.  The default, 0, runs the
# worker thread in the current process.  Greater than 0 launches that number
 of
# child processes as workers.  The parent process manages them.
# api_workers = 0

# Number of separate RPC worker processes to spawn.  The default, 0, runs the
# 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

# Timeout for client connections socket operations. If an
# incoming connection is idle for this number of seconds it
# will be closed. A value of '0' means wait forever. (integer
# value)
# client_socket_timeout = 900

# 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 =

# The name of the admin nova tenant. If the uuid of the admin nova tenant
# is set, this is optional.  Useful for cases where the uuid of the admin
# nova tenant is not available when configuration is being done.
# nova_admin_tenant_name =

# Password for connection to nova in admin context.
# nova_admin_password =

# Authorization URL for connection to nova in admin context.
# nova_admin_auth_url =

# CA file for novaclient to verify server certificates
# nova_ca_certificates_file =

# Boolean to control ignoring SSL errors on the nova url
# nova_api_insecure = False

# Number of seconds between sending events to nova if there are any events to
 send
# send_events_interval = 2

# ======== end of neutron nova interactions ==========

#
# 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 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

# SSL version to use (valid only if SSL enabled). valid values
# are TLSv1, SSLv23 and SSLv3. SSLv2 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=AMQPLAIN

# 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

# 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=

# 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=oslo

# 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=

# 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


[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=


[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

[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.
# quota_health_monitor = -1

# Number of loadbalancers allowed per tenant. A negative value means
 unlimited.
# quota_loadbalancer = 10

# Number of listeners allowed per tenant. A negative value means unlimited.
# quota_listener = -1

# Number of v2 health monitors allowed per tenant. A negative value means
# unlimited. These health monitors exist under the lbaas v2 API
# quota_healthmonitor = -1

# Number of routers allowed per tenant. A negative value means unlimited.
# quota_router = 10

# Number of floating IPs allowed per tenant. A negative value means unlimited.
# quota_floatingip = 50

# Number of firewalls allowed per tenant. A negative value means unlimited.
# quota_firewall = 1

# Number of firewall policies allowed per tenant. A negative value means
# unlimited.
# quota_firewall_policy = 1

# Number of firewall rules allowed per tenant. A negative value means
# unlimited.
# quota_firewall_rule = 100

[agent]
# Use "sudo neutron-rootwrap /etc/neutron/rootwrap.conf" to use the real
# root filter facility.
# Change to "sudo" to skip the filtering and just run the command directly
# root_helper = sudo

# Set to true to add comments to generated iptables rules that describe
# each rule's purpose. (System must support the iptables comments module.)
# comment_iptables_rules = True

# Root helper daemon application to use when possible.
# root_helper_daemon =

# Use the root helper when listing the namespaces on a system. This may not
# be required depending on the security configuration. If the root helper is
# not required, set this to False for a performance improvement.
# use_helper_for_ns_read = True

# The interval to check external processes for failure in seconds (0=disabled)
# check_child_processes_interval = 60

# Action to take when an external process spawned by an agent dies
# Values:
#   respawn - Respawns the external process
#   exit - Exits the agent
# check_child_processes_action = respawn
```

```
# ========== items for agent management extension =============
# 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
# report_interval = 30

# ==========  end of items for agent management extension =====

[keystone_authtoken]
auth_uri = http://127.0.0.1:35357/v2.0/
identity_uri = http://127.0.0.1:5000
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%

[database]
# This line MUST be changed to actually run the plugin.
# Example:
# connection = mysql://root:pass@127.0.0.1:3306/neutron
# Replace 127.0.0.1 above with the IP address of the database used by the
# main neutron server. (Leave it as is if the database runs on this host.)
# connection = sqlite://
# NOTE: In deployment the [database] section and its connection attribute may
# be set in the corresponding core plugin '.ini' file. However, it is
 suggested
# to put the [database] section and its connection attribute in this
# configuration file.

# Database engine for which script will be generated when using offline
# migration
# engine =

# The SQLAlchemy connection string used to connect to the slave database
# slave_connection =

# Database reconnection retry times - in event connectivity is lost
# set to -1 implies an infinite retry count
# max_retries = 10

# Database reconnection interval in seconds - if the initial connection to the
# database fails
# retry_interval = 10

# Minimum number of SQL connections to keep open in a pool
# min_pool_size = 1

# Maximum number of SQL connections to keep open in a pool
# max_pool_size = 10

# Timeout in seconds before idle sql connections are reaped
# idle_timeout = 3600

# If set, use this value for max_overflow with sqlalchemy
# max_overflow = 20

# 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

[nova]
# Name of the plugin to load
# auth_plugin =

# Config Section from which to load plugin specific options
# auth_section =

# PEM encoded Certificate Authority to use when verifying HTTPs connections.
# cafile =

# PEM encoded client certificate cert file
# certfile =

# Verify HTTPS connections.
# insecure = False

# PEM encoded client certificate key file
# keyfile =

# Name of nova region to use. Useful if keystone manages more than one region.
# region_name =

# Timeout value for http requests
# timeout =

[oslo_concurrency]

# Directory to use for lock files. For security, the specified directory
 should
# only be writable by the user running the processes that need locking.
# Defaults to environment variable OSLO_LOCK_PATH. If external locks are used,
# a lock path must be set.
lock_path = $state_path/lock

# Enables or disables inter-process locks.
# disable_process_locking = False

[oslo_policy]

# The JSON file that defines policies.
# policy_file = policy.json

# Default rule. Enforced when a requested rule is not found.
# policy_default_rule = default

# Directories where policy configuration files are stored.
# They can be relative to any directory in the search path defined by the
# config_dir option, or absolute paths. The file defined by policy_file
# must exist for these directories to be searched. Missing or empty
# directories are ignored.
# policy_dirs = policy.d

[oslo_messaging_amqp]

#
# From oslo.messaging
#
```

```
# Address prefix used when sending to a specific server (string value)
# Deprecated group/name - [amqp1]/server_request_prefix
# server_request_prefix = exclusive

# Address prefix used when broadcasting to all servers (string value)
# Deprecated group/name - [amqp1]/broadcast_prefix
# broadcast_prefix = broadcast

# Address prefix when sending to any server in group (string value)
# Deprecated group/name - [amqp1]/group_request_prefix
# group_request_prefix = unicast

# Name for the AMQP container (string value)
# Deprecated group/name - [amqp1]/container_name
# container_name =

# Timeout for inactive connections (in seconds) (integer value)
# Deprecated group/name - [amqp1]/idle_timeout
# idle_timeout = 0

# Debug: dump AMQP frames to stdout (boolean value)
# Deprecated group/name - [amqp1]/trace
# trace = false

# CA certificate PEM file for verifing server certificate (string value)
# Deprecated group/name - [amqp1]/ssl_ca_file
# ssl_ca_file =

# Identifying certificate PEM file to present to clients (string value)
# Deprecated group/name - [amqp1]/ssl_cert_file
# ssl_cert_file =

# Private key PEM file used to sign cert_file certificate (string value)
# Deprecated group/name - [amqp1]/ssl_key_file
# ssl_key_file =

# Password for decrypting ssl_key_file (if encrypted) (string value)
# Deprecated group/name - [amqp1]/ssl_key_password
# ssl_key_password =

# Accept clients using either SSL or plain TCP (boolean value)
# Deprecated group/name - [amqp1]/allow_insecure_clients
# allow_insecure_clients = false


[oslo_messaging_qpid]

#
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
# amqp_auto_delete = false
```

```
# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
# rpc_conn_pool_size = 30

# Qpid broker hostname. (string value)
# Deprecated group/name - [DEFAULT]/qpid_hostname
# qpid_hostname = localhost

# Qpid broker port. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_port
# qpid_port = 5672

# Qpid HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/qpid_hosts
# qpid_hosts = $qpid_hostname:$qpid_port

# Username for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_username
# qpid_username =

# Password for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_password
# qpid_password =

# Space separated list of SASL mechanisms to use for auth. (string value)
# Deprecated group/name - [DEFAULT]/qpid_sasl_mechanisms
# qpid_sasl_mechanisms =

# Seconds between connection keepalive heartbeats. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_heartbeat
# qpid_heartbeat = 60

# Transport to use, either 'tcp' or 'ssl'. (string value)
# Deprecated group/name - [DEFAULT]/qpid_protocol
# qpid_protocol = tcp

# Whether to disable the Nagle algorithm. (boolean value)
# Deprecated group/name - [DEFAULT]/qpid_tcp_nodelay
# qpid_tcp_nodelay = true

# The number of prefetched messages held by receiver. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_receiver_capacity
# 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)
# Deprecated group/name - [DEFAULT]/qpid_topology_version
# qpid_topology_version = 1


[oslo_messaging_rabbit]

#
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
# amqp_auto_delete = false

# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
# rpc_conn_pool_size = 30

# SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and
# SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some
# distributions. (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_version
# kombu_ssl_version =

# SSL key file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_keyfile
# kombu_ssl_keyfile =

# SSL cert file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_certfile
# kombu_ssl_certfile =

# SSL certification authority file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_ca_certs
# kombu_ssl_ca_certs =

# How long to wait before reconnecting in response to an AMQP consumer cancel
# notification. (floating point value)
# Deprecated group/name - [DEFAULT]/kombu_reconnect_delay
# kombu_reconnect_delay = 1.0

# The RabbitMQ broker address where a single node is used. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_host
# rabbit_host = localhost

# The RabbitMQ broker port where a single node is used. (integer value)
# Deprecated group/name - [DEFAULT]/rabbit_port
# rabbit_port = 5672

# RabbitMQ HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/rabbit_hosts
# rabbit_hosts = $rabbit_host:$rabbit_port

# Connect over SSL for RabbitMQ. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_use_ssl
# rabbit_use_ssl = false

# The RabbitMQ userid. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_userid
# rabbit_userid = guest

# The RabbitMQ password. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_password
# rabbit_password = guest

# The RabbitMQ login method. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_login_method
```

```
# rabbit_login_method = AMQPLAIN

# The RabbitMQ virtual host. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_virtual_host
# 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)
# Deprecated group/name - [DEFAULT]/rabbit_retry_backoff
# rabbit_retry_backoff = 2

# Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry
# count). (integer value)
# Deprecated group/name - [DEFAULT]/rabbit_max_retries
# 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)
# Deprecated group/name - [DEFAULT]/rabbit_ha_queues
# rabbit_ha_queues = false

# Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake (boolean value)
# Deprecated group/name - [DEFAULT]/fake_rabbit
# fake_rabbit = false
```

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 = oslo.middleware:RequestId.factory

[filter:catch_errors]
paste.filter_factory = oslo.middleware:CatchErrors.factory

[filter:keystonecontext]
paste.filter_factory = neutron.auth:NeutronKeystoneContext.factory

[filter:authtoken]
paste.filter_factory = keystonemiddleware.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",
    "context_is_advsvc":  "role:advsvc",
    "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",
    "shared_firewall_policies": "field:firewall_policies:shared=True",
    "shared_subnetpools": "field:subnetpools:shared=True",
    "external": "field:networks:router:external=True",
    "default": "rule:admin_or_owner",

    "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_subnetpool": "",
    "create_subnetpool:shared": "rule:admin_only",
    "get_subnetpool": "rule:admin_or_owner or rule:shared_subnetpools",
    "update_subnetpool": "rule:admin_or_owner",
    "delete_subnetpool": "rule:admin_or_owner",

    "create_network": "",
    "get_network": "rule:admin_or_owner or rule:shared or rule:external or
 rule:context_is_advsvc",
    "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",
    "update_network:router:external": "rule:admin_only",
    "delete_network": "rule:admin_or_owner",

    "create_port": "",
    "create_port:mac_address": "rule:admin_or_network_owner or
rule:context_is_advsvc",
    "create_port:fixed_ips": "rule:admin_or_network_owner or
rule:context_is_advsvc",
    "create_port:port_security_enabled": "rule:admin_or_network_owner or
rule:context_is_advsvc",
    "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 or
rule:context_is_advsvc",
    "create_port:allowed_address_pairs": "rule:admin_or_network_owner",
    "get_port": "rule:admin_or_owner or rule:context_is_advsvc",
    "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 or rule:context_is_advsvc",
    "update_port:mac_address": "rule:admin_only or rule:context_is_advsvc",
    "update_port:fixed_ips": "rule:admin_or_network_owner or
rule:context_is_advsvc",
    "update_port:port_security_enabled": "rule:admin_or_network_owner or
rule:context_is_advsvc",
    "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 or
rule:context_is_advsvc",
    "update_port:allowed_address_pairs": "rule:admin_or_network_owner",
    "delete_port": "rule:admin_or_owner or rule:context_is_advsvc",

    "get_router:ha": "rule:admin_only",
    "create_router": "rule:regular_user",
    "create_router:external_gateway_info:enable_snat": "rule:admin_only",
    "create_router:distributed": "rule:admin_only",
    "create_router:ha": "rule:admin_only",
    "get_router": "rule:admin_or_owner",
    "get_router:distributed": "rule:admin_only",
    "update_router:external_gateway_info:enable_snat": "rule:admin_only",
    "update_router:distributed": "rule:admin_only",
    "update_router:ha": "rule:admin_only",
    "delete_router": "rule:admin_or_owner",

    "add_router_interface": "rule:admin_or_owner",
    "remove_router_interface": "rule:admin_or_owner",

    "create_router:external_gateway_info:external_fixed_ips":
"rule:admin_only",
    "update_router:external_gateway_info:external_fixed_ips":
"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",
    "update_firewall:shared": "rule:admin_only",
    "delete_firewall": "rule:admin_or_owner",

    "create_firewall_policy": "",
    "get_firewall_policy": "rule:admin_or_owner or
 rule:shared_firewall_policies",
    "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": "",
    "get_firewall_rule": "rule:admin_or_owner or rule:shared_firewalls",
    "update_firewall_rule": "rule:admin_or_owner",
    "delete_firewall_rule": "rule:admin_or_owner",

    "create_qos_queue": "rule:admin_only",
    "get_qos_queue": "rule:admin_only",

    "update_agent": "rule:admin_only",
    "delete_agent": "rule:admin_only",
    "get_agent": "rule:admin_only",

    "create_dhcp-network": "rule:admin_only",
    "delete_dhcp-network": "rule:admin_only",
    "get_dhcp-networks": "rule:admin_only",
    "create_l3-router": "rule:admin_only",
    "delete_l3-router": "rule:admin_only",
    "get_l3-routers": "rule:admin_only",
    "get_dhcp-agents": "rule:admin_only",
    "get_l3-agents": "rule:admin_only",
    "get_loadbalancer-agent": "rule:admin_only",
    "get_loadbalancer-pools": "rule:admin_only",
    "get_agent-loadbalancers": "rule:admin_only",
    "get_loadbalancer-hosting-agent": "rule:admin_only",

    "create_floatingip": "rule:regular_user",
    "create_floatingip:floating_ip_address": "rule:admin_only",
    "update_floatingip": "rule:admin_or_owner",
    "delete_floatingip": "rule:admin_or_owner",
    "get_floatingip": "rule:admin_or_owner",

    "create_network_profile": "rule:admin_only",
    "update_network_profile": "rule:admin_only",
    "delete_network_profile": "rule:admin_only",
    "get_network_profiles": "",
    "get_network_profile": "",
    "update_policy_profiles": "rule:admin_only",
    "get_policy_profiles": "",
    "get_policy_profile": "",

    "create_metering_label": "rule:admin_only",
    "delete_metering_label": "rule:admin_only",
    "get_metering_label": "rule:admin_only",

    "create_metering_label_rule": "rule:admin_only",
    "delete_metering_label_rule": "rule:admin_only",
```

```
    "get_metering_label_rule": "rule:admin_only",

    "get_service_provider": "rule:regular_user",
    "get_lsn": "rule:admin_only",
    "create_lsn": "rule:admin_only"
}
```

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

# List of directories to search executables in, in case filters do not
# explicitely 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). This option is deprecated and
# will be removed in a future release, at which point the old behavior of
# use_namespaces = True will be enforced.
# 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

# Use broadcast in DHCP replies
# dhcp_broadcast_reply = False

# 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). This option is deprecated and
# will be removed in a future release, at which point the old behavior of
# use_namespaces = True will be enforced.
# 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 =

# With IPv6, the network used for the external gateway does not need
# to have an associated subnet, since the automatically assigned
# link-local address (LLA) can be used. However, an IPv6 gateway address
# is needed for use as the next-hop for the default route. If no IPv6
# gateway address is configured here, (and only then) the neutron router
# will be configured to get its default route from router advertisements (RAs)
# from the upstream router; in which case the upstream router must also be
# configured to send these RAs.
# The ipv6_gateway, when configured, should be the LLA of the interface
# on the upstream router. If a next-hop using a global unique address (GUA)
# is desired, it needs to be done via a subnet allocated to the network
# and not through this parameter.
# ipv6_gateway =

# 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 = 3

# 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
```

```
# Iptables mangle mark used to mark metadata valid requests
# metadata_access_mark = 0x1

# Iptables mangle mark used to mark ingress from external network
# external_ingress_mark = 0x2

# 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

# The working mode for the agent. Allowed values are:
# - legacy: this preserves the existing behavior where the L3 agent is
#   deployed on a centralized networking node to provide L3 services
#   like DNAT, and SNAT. Use this mode if you do not want to adopt DVR.
# - dvr: this mode enables DVR functionality, and must be used for an L3
#   agent that runs on a compute host.
# - dvr_snat: this enables centralized SNAT support in conjunction with
#   DVR. This mode must be used for an L3 agent running on a centralized
#   node (or in single-host deployments, e.g. devstack).
# agent_mode = legacy

# Location to store keepalived and all HA configurations
# ha_confs_path = $state_path/ha_confs

# VRRP authentication type AH/PASS
# ha_vrrp_auth_type = PASS

# VRRP authentication password
# ha_vrrp_auth_password =

# The advertisement interval in seconds
# ha_vrrp_advert_int = 2
```

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

# Which protocol to use for requests to Nova metadata server, http or https
# nova_metadata_protocol = http

# Whether insecure SSL connection should be accepted for Nova metadata server
# requests
# nova_metadata_insecure = False

# Client certificate for nova api, needed when nova api requires client
# certificates
# nova_client_cert =

# Private key for nova client certificate
# nova_client_priv_key =

# 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 the same config key, but in [neutron] section.
# metadata_proxy_shared_secret =

# Location of Metadata Proxy UNIX domain socket
# metadata_proxy_socket = $state_path/metadata_proxy

# Metadata Proxy UNIX domain socket mode, 3 values allowed:
# 'deduce': deduce mode from metadata_proxy_user/group values,
# 'user': set metadata proxy socket mode to 0o644, to use when
# metadata_proxy_user is agent effective user or root,
# 'group': set metadata proxy socket mode to 0o664, to use when
# metadata_proxy_group is agent effective group,
# 'all': set metadata proxy socket mode to 0o666, to use otherwise.
# metadata_proxy_socket_mode = deduce

# Number of separate worker processes for metadata server. Defaults to
# half the number of CPU cores
# metadata_workers =

# Number of backlog requests to configure the metadata server socket with
# metadata_backlog = 4096

# URL to connect to the cache backend.
# 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 = memory://?default_ttl=5
```

New, updated and deprecated options in Kilo for OpenStack Networking

Table 9.68. New options

Option = default value	(Type) Help string
[DEFAULT] advertise_mtu = False	(BoolOpt) If True, effort is made to advertise MTU settings to VMs via network methods (DHCP and RA MTU options) when the network's preferred MTU is known.
[DEFAULT] allow_automatic_dhcp_failover = True	(BoolOpt) Automatically remove networks from offline DHCP agents.
[DEFAULT] conn_idle_timeout = 900	(IntOpt) Reconnect connection to nsx if not used within this amount of time.
[DEFAULT] default_ipv4_subnet_pool = None	(StrOpt) Default IPv4 subnet-pool to be used for automatic subnet CIDR allocation
[DEFAULT] default_ipv6_subnet_pool = None	(StrOpt) Default IPv6 subnet-pool to be used for automatic subnet CIDR allocation
[DEFAULT] dhcp_broadcast_reply = False	(BoolOpt) Use broadcast in DHCP replies
[DEFAULT] dhcp_load_type = networks	(StrOpt) Representing the resource type whose load is being reported by the agent. This can be "networks", "subnets" or "ports". When specified (Default is networks), the server will extract particular load sent as part of its agent configuration object from the agent report state, which is the number of resources being consumed, at every report_interval.dhcp_load_type can be used in combination with network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.WeightScheduler When the network_scheduler_driver is WeightScheduler, dhcp_load_type can be configured to represent the choice for the resource being balanced. Example: dhcp_load_type=networks
[DEFAULT] enable_services_on_agents_with_admin_state_down = False	(BoolOpt) Enable services on an agent with admin_state_up False. If this option is False, when admin_state_up of an agent is turned False, services on it will be disabled. Agents with admin_state_up False are not selected for automatic scheduling regardless of this option. But manual scheduling to such agents is available if this option is True.
[DEFAULT] external_ingress_mark = 0x2	(StrOpt) Iptables mangle mark used to mark ingress from external network
[DEFAULT] ipv6_gateway =	(StrOpt) With IPv6, the network used for the external gateway does not need to have an associated subnet, since the automatically assigned link-local address (LLA) can be used. However, an IPv6 gateway address is needed for use as the next-hop for the default route. If no IPv6 gateway address is configured here, (and only then) the neutron router will be configured to get its default route from router advertisements (RAs) from the upstream router; in which case the upstream router must also be configured to send these RAs. The ipv6_gateway, when configured, should be the LLA of the interface on the upstream router. If a next-hop using a global unique address (GUA) is desired, it needs to be done via a subnet allocated to the network and not through this parameter.
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.

Option = default value	(Type) Help string
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] metadata_access_mark = 0x1	(StrOpt) Iptables mangle mark used to mark metadata valid requests
[DEFAULT] metadata_proxy_group =	(StrOpt) Group (gid or name) running metadata proxy after its initialization (if empty: agent effective group).
[DEFAULT] metadata_proxy_socket_mode = deduce	(StrOpt) Metadata Proxy UNIX domain socket mode, 3 values allowed: 'deduce': deduce mode from metadata_proxy_user/group values, 'user': set metadata proxy socket mode to 0o644, to use when metadata_proxy_user is agent effective user or root, 'group': set metadata proxy socket mode to 0o664, to use when metadata_proxy_group is agent effective group or root, 'all': set metadata proxy socket mode to 0o666, to use otherwise.
[DEFAULT] metadata_proxy_user =	(StrOpt) User (uid or name) running metadata proxy after its initialization (if empty: agent effective user).
[DEFAULT] metadata_proxy_watch_log = None	(BoolOpt) Enable/Disable log watch by metadata proxy. It should be disabled when metadata_proxy_user/group is not allowed to read/write its log file and copytruncate logrotate option must be used if logrotate is enabled on metadata proxy log files. Option default value is deduced from metadata_proxy_user: watch log is enabled if metadata_proxy_user is agent effective user id/name.
[DEFAULT] nova_admin_tenant_name = None	(StrOpt) The name of the admin nova tenant. Deprecated in favour of an auth plugin in [nova].
[DEFAULT] policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
[DEFAULT] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
[DEFAULT] vlan_transparent = False	(BoolOpt) If True, then allow plugins that support it to create VLAN transparent networks.
[AGENT] check_child_processes_action = respawn	(StrOpt) Action to be executed when a child process dies
[AGENT] check_child_processes_interval = 60	(IntOpt) Interval between checks of child process liveness (seconds), use 0 to disable

Option = default value	(Type) Help string
[AGENT] comment_iptables_rules = True	(BoolOpt) Add comments to iptables rules.
[AGENT] prevent_arp_spoofing = False	(BoolOpt) Enable suppression of ARP responses that don't match an IP address that belongs to the port from which they originate. Note: This prevents the VMs attached to this agent from spoofing, it doesn't protect them from other devices which have the capability to spoof (e.g. bare metal or VMs attached to agents without this flag set to True). Spoofing rules will not be added to any ports that have port security disabled. This requires a version of OVS that supports matching ARP headers.
[AGENT] root_helper_daemon = None	(StrOpt) Root helper daemon application to use when possible.
[AGENT] use_helper_for_ns_read = True	(BoolOpt) Use the root helper to read the namespaces from the operating system.
[MIDONET] client = midonet.neutron.client.api.MidonetApiClient	(StrOpt) MidoNet client used to access MidoNet data storage.
[MIDONET] cluster_ip = localhost	(StrOpt) IP that the cluster service can be reached on
[MIDONET] cluster_port = 8088	(StrOpt) Port that the cluster service can be reached on
[MIDONET] tunnel_protocol = vxlan	(StrOpt) Tunnel protocol used by Midonet
[OFC] support_packet_filter_on_ofc_router = True	(BoolOpt) Support packet filter on OFC router interface.
[OVSVAPP] agent_driver = networking_vsphere.agent.ovsvapp_agent.OVSvAppL2Agent	(StrOpt) OVSvApp Agent implementation.
[OVSVAPP] agent_maintenance = False	(BoolOpt) Turn on this flag during agent updates to help prevent datapath outage.
[OVSVAPP] bridge_mappings =	(ListOpt) Bridge mappings.
[OVSVAPP] dont_fragment = True	(IntOpt) Do not fragment.
[OVSVAPP] integration_bridge = br-int	(StrOpt) Integration Bridge.
[OVSVAPP] local_ip =	(StrOpt) Local IP address of VXLAN tunnel endpoint.
[OVSVAPP] network_manager = networking_vsphere.drivers.manager.VcenterManager	(StrOpt) Driver Manager implementation for NetworkDriver.
[OVSVAPP] polling_interval = 2	(IntOpt) The number of seconds the agent will wait between polling for local device changes.
[OVSVAPP] report_interval = 4	(IntOpt) Seconds between nodes reporting state to server.
[OVSVAPP] tenant_network_type = vlan	(StrOpt) Network type for tenant networks
[OVSVAPP] tunnel_bridge = br-tun	(StrOpt) Tunnel Bridge for tunneling.
[OVSVAPP] tunnel_types = vxlan	(ListOpt) Tunnel network types supported by the OVSvApp Agent.
[OVSVAPP] veth_mtu = 1500	(IntOpt) MTU size of veth interfaces.
[OVSVAPP] vxlan_udp_port = 4789	(IntOpt) The UDP port to use for VXLAN tunnels.
[SECURITYGROUP] defer_apply = True	(BoolOpt) Enable defer_apply on security bridge.
[SECURITYGROUP] ovsvapp_firewall_driver = networking_vsphere.drivers.ovs_firewall.OVSFirewallDriver	(StrOpt) DriverManager implementation for OVS based Firewall.
[SECURITYGROUP] security_bridge_mapping = br-sec	(StrOpt) <security_bridge>:<phy_interface>
[VMWARE] cert_check = False	(BoolOpt) Enable SSL certificate check for vCenter.
[VMWARE] cert_path = /etc/ssl/certs/certs.pem	(StrOpt) Certificate chain path containing cacert of vCenters.
[VMWARE] cluster_dvs_mapping = ['']	(MultiStrOpt) vCenter cluster to DVS mapping.
[VMWARE] esx_hostname = None	(StrOpt) ESX host name where this OVSvApp is hosted.
[VMWARE] esx_maintenance_mode = True	(BoolOpt) Set host into maintenance mode.
[VMWARE] https_port = 443	(IntOpt) Customized https_port for vCenter communication.

Option = default value	(Type) Help string
[VMWARE] vcenter_api_retry_count = 5	(StrOpt) Number of retries while connecting to vcenter server.
[VMWARE] vcenter_id = None	(StrOpt) Unique ID of the vCenter Server on which this OVSvApp ishosted
[VMWARE] vcenter_ip = None	(StrOpt) vCenter server IP.
[VMWARE] vcenter_password = None	(StrOpt) vCenter server password.
[VMWARE] vcenter_username = None	(StrOpt) vCenter server user name.
[VMWARE] wsdl_location = None	(StrOpt) vCenter server wsdl location.
[VROUTER] flavor = 2	(StrOpt) Nova VM flavor for instances of Vyatta vRouter.
[VROUTER] image_id = None	(StrOpt) Nova image id for instances of Vyatta vRouter.
[VROUTER] keystone_url = None	(StrOpt) Keystone URL.
[VROUTER] management_network_id = None	(StrOpt) Vyatta vRouter management network id.
[VROUTER] nova_poll_interval = 5	(IntOpt) Number of seconds between consecutive Nova queries when waiting for router instance status change.
[VROUTER] nova_spawn_timeout = 300	(IntOpt) Number of seconds to wait for Nova to activate instance before setting resource to error state.
[VROUTER] tenant_admin_name = None	(StrOpt) Name of tenant admin user.
[VROUTER] tenant_admin_password = None	(StrOpt) Tenant admin password.
[VROUTER] tenant_id = None	(StrOpt) UUID of tenant that holds Vyatta vRouter instances.
[VROUTER] vrouter_boot_timeout = 300	(IntOpt) Number of seconds to wait for Vyatta vRouter to boot before setting resource to error state.
[VROUTER] vrouter_credentials = vyatta:vyatta	(StrOpt) Vyatta vRouter login credentials
[VROUTER] vrouter_poll_interval = 5	(IntOpt) Number of seconds between consecutive Vyatta vRouter queries when waiting for router instance boot.
[l3_brocade_mlx] switch_names =	(StrOpt) Switches connected to the compute nodes
[ml2] path_mtu = 0	(IntOpt) The maximum permissible size of an unfragmented packet travelling from and to addresses where encapsulated Neutron traffic is sent. If <= 0, the path MTU is indeterminate.
[ml2] physical_network_mtus =	(ListOpt) A list of mappings of physical networks to MTU values. The format of the mapping is <physnet>:<mtu val>. This mapping allows specifying a physical network MTU value that differs from the default segment_mtu value.
[ml2] segment_mtu = 0	(IntOpt) The maximum permissible size of an unfragmented packet travelling a L2 network segment. If <= 0, the segment MTU is indeterminate.
[ml2_brocade] rbridge_id = 1	(StrOpt) Rbridge id of provider edge router(s)
[ml2_brocade_fi_ni] switch_names =	(StrOpt) Switches connected to the compute nodes
[ml2_cisco] host_key_checks = False	(BoolOpt) Enable strict host key checks when connecting to Nexus switches
[ml2_cisco] persistent_switch_config = False	(BoolOpt) To make Nexus configuration persistent
[ml2_cisco] provider_vlan_auto_create = True	(BoolOpt) Provider VLANs are automatically created as needed on the Nexus switch
[ml2_cisco] provider_vlan_auto_trunk = True	(BoolOpt) Provider VLANs are automatically trunked as needed on the ports of the Nexus switch
[ml2_cisco] provider_vlan_name_prefix = p-	(StrOpt) VLAN Name prefix for provider vlans
[ml2_cisco] switch_heartbeat_time = 0	(IntOpt) Periodic time to check switch connection. (0=disabled)
[ml2_cisco] switch_replay_count = 3	(IntOpt) Number of times to attempt config replay with switch.

Option = default value	(Type) Help string
[ml2_cisco] vxlan_global_config = False	(BoolOpt) Create and delete Nexus switch VXLAN global settings; feature nv overlay, feature vn-segment-vlan-based, interface nve + source-interface loopback
[ml2_cisco_n1kv] default_policy_profile = default-pp	(StrOpt) Cisco Nexus1000V default policy profile.
[ml2_cisco_n1kv] default_vlan_network_profile = default-vlan-np	(StrOpt) Cisco Nexus1000V default network profile for VLAN networks.
[ml2_cisco_n1kv] default_vxlan_network_profile = default-vxlan-np	(StrOpt) Cisco Nexus1000V default network profile for VXLAN networks.
[ml2_cisco_n1kv] http_pool_size = 4	(IntOpt) Number of threads to use to make HTTP requests.
[ml2_cisco_n1kv] http_timeout = 15	(IntOpt) HTTP timeout, in seconds, for connections to the Cisco Nexus1000V VSMs.
[ml2_cisco_n1kv] n1kv_vsm_ips = None	(ListOpt) Comma Separated IP Addresses of the Cisco Nexus1000V VSMs.
[ml2_cisco_n1kv] password = None	(StrOpt) Password for all configured Cisco Nexus1000V VSMs.
[ml2_cisco_n1kv] poll_duration = 60	(IntOpt) Cisco Nexus1000V policy profile polling duration in seconds.
[ml2_cisco_n1kv] restrict_policy_profiles = False	(BoolOpt) Restrict the visibility of policy profiles to the tenants.
[ml2_cisco_n1kv] sync_interval = 300	(IntOpt) Time interval between consecutive neutron-VSM syncs.
[ml2_cisco_n1kv] username = None	(StrOpt) Username for all configured Cisco Nexus1000V VSMs.
[ml2_cisco_ucsm] supported_pci_devs = 1137:0071, 8086:10c9	(ListOpt) List of comma separated vendor_id:product_id of SR_IOV capable devices supported by this MD. This MD supports both VM-FEX and SR-IOV devices.
[ml2_cisco_ucsm] ucsm_host_list = None	(ListOpt) List of comma separated Host:Service Profile tuples providing the Service Profile associated with each Host to be supported by this MD.
[ml2_cisco_ucsm] ucsm_ip = None	(StrOpt) Cisco UCS Manager IP address. This is a required field to communicate with a Cisco UCS Manager.
[ml2_cisco_ucsm] ucsm_password = None	(StrOpt) Password for UCS Manager. This is a required field to communicate with a Cisco UCS Manager.
[ml2_cisco_ucsm] ucsm_username = None	(StrOpt) Username for UCS Manager. This is a required field to communicate with a Cisco UCS Manager.
[nova] auth_plugin = None	(StrOpt) Name of the plugin to load
[nova] auth_section = None	(StrOpt) Config Section from which to load plugin specific options
[nova] cafile = None	(StrOpt) PEM encoded Certificate Authority to use when verifying HTTPs connections.
[nova] certfile = None	(StrOpt) PEM encoded client certificate cert file
[nova] insecure = False	(BoolOpt) Verify HTTPS connections.
[nova] keyfile = None	(StrOpt) PEM encoded client certificate key file
[nova] region_name = None	(StrOpt) Name of nova region to use. Useful if keystone manages more than one region.
[nova] timeout = None	(IntOpt) Timeout value for http requests
[nsx_v3] default_edge_cluster_uuid = None	(StrOpt) Default edge cluster identifier
[nsxv] backup_edge_pool = service:large:4:10, service:compact:4:10, vdr:large:4:10	(ListOpt) Defines edge pool using the format: <edge_type>:[edge_size]:<min_edges>:<max_edges>.edge_type: service,vdr. edge_size: compact, large, xlarge, quadlarge and default is large.

Option = default value	(Type) Help string
[nsxv] cluster_moid =	(ListOpt) Parameter listing the IDs of the clusters which are used by OpenStack.
[nsxv] datacenter_moid = None	(StrOpt) Optional parameter identifying the ID of datacenter to deploy NSX Edges
[nsxv] datastore_id = None	(StrOpt) Optional parameter identifying the ID of datastore to deploy NSX Edges
[nsxv] deployment_container_id = None	(StrOpt) Optional parameter identifying the ID of datastore to deploy NSX Edges
[nsxv] dvs_id = None	(StrOpt) DVS ID for VLANs
[nsxv] external_network = None	(StrOpt) Network ID for physical network connectivity
[nsxv] manager_uri = None	(StrOpt) uri for vsm
[nsxv] maximum_tunnels_per_vnic = 20	(IntOpt) Maximum number of sub interfaces supported per vnic in edge. The value should be in 1-110.
[nsxv] metadata_shared_secret = None	(StrOpt) Shared secret to sign metadata requests
[nsxv] mgt_net_default_gateway = None	(StrOpt) Management network default gateway for metadata proxy
[nsxv] mgt_net_moid = None	(StrOpt) Network ID for management network connectivity
[nsxv] mgt_net_proxy_ips = None	(ListOpt) Management network IP address for metadata proxy
[nsxv] mgt_net_proxy_netmask = None	(StrOpt) Management network netmask for metadata proxy
[nsxv] nova_metadata_ips = None	(ListOpt) IP addresses used by Nova metadata service
[nsxv] nova_metadata_port = 8775	(IntOpt) TCP Port used by Nova metadata server
[nsxv] password = default	(StrOpt) Password for vsm
[nsxv] resource_pool_id = None	(StrOpt) Optional parameter identifying the ID of resource to deploy NSX Edges
[nsxv] retries = 10	(IntOpt) Maximum number of API retries on endpoint.
[nsxv] spoofguard_enabled = True	(BoolOpt) If True then plugin will use NSXV spoofguard component for port-security feature.
[nsxv] task_status_check_interval = 2000	(IntOpt) Task status check interval
[nsxv] tenant_router_types = shared, distributed, exclusive	(ListOpt) Ordered list of router_types to allocate as tenant routers.
[nsxv] user = admin	(StrOpt) User name for vsm
[nsxv] vdn_scope_id = None	(StrOpt) Network scope ID for VXLAN virtual wires
[oslo_concurrency] disable_process_locking = False	(BoolOpt) Enables or disables inter-process locks.
[oslo_concurrency] lock_path = None	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate

Option = default value	(Type) Help string
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.

Option = default value	(Type) Help string
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.

Table 9.69. New default values

Option	Previous default value	New default value
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	local
[CISCO_N1K] restrict_network_profiles	False	True
[NOVA] vif_type	ovs	ivs
[OVS] local_ip		None
[OVS] ovsdb_interface	None	vsctl
[RESTPROXYAGENT] virtual_switch_type	ovs	ivs
[VXLAN] enable_vxlan	False	True
[VXLAN] local_ip		None
[hosting_devices] csr1kv_cfgagent_router_driver	neutron.plugins.cisco.cfg_agent.device_drivers.csr1kv.csr1kv_routing_driver.CSR1kvRoutingDriver	networking_cisco.plugins.cisco.cfg_agent.device_drivers.csr1kv.csr1kv_routing_driver.CSR1kvRoutingDriver
[hosting_devices] csr1kv_device_driver	neutron.plugins.cisco.l3.hosting_device_drivers.csr1kv_hd_driver.CSR1kvHostingDeviceDriver	networking_cisco.plugins.cisco.l3.hosting_device_drivers.csr1kv_hd_driver.CSR1kvHostingDeviceDriver

Option	Previous default value	New default value
[hosting_devices] csr1kv_plugging_driver	neutron.plugins.cisco.l3.plugging_drivers.n1kv_trunking_driver.N1kvTrunkingPlugDriver	neutron.plugins.cisco.l3.plugging_drivers.n1kv_trun
[ml2_sriov] supported_pci_vendor_devs	15b3:1004, 8086:10c9	15b3:1004, 8086:10ca
[plumgriddirector] driver	neutron.plugins.plumgrid.drivers.plumlib.Plumlib	networking_plumgrid.neutron.plugins.drivers.plumlib.Plumlib

Table 9.70. Deprecated options

Deprecated option	New Option
[DEFAULT] use-syslog	None
[DEFAULT] use_syslog	None
[DEFAULT] max_request_body_size	[oslo_middleware] max_request_body_size
[DEFAULT] nova_api_insecure	[nova] insecure
[DEFAULT] log-format	None
[DEFAULT] nova_ca_certificates_file	[nova] cafile
[DEFAULT] nova_region_name	[nova] region_name
[DEFAULT] log_format	None

10. Object Storage

Table of Contents

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 10.1. Description of configuration options for `[swift-hash]` in `swift.conf`

Configuration option = Default value	Description
`swift_hash_path_prefix` = *changeme*	A prefix used by hash_path 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 ouside 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.
`swift_hash_path_suffix` = *changeme*	A suffix used by hash_path 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 ouside 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 10.2. Description of configuration options for [DEFAULT] in object-server.conf

Configuration option = Default value	Description
backlog = *4096*	Maximum number of allowed pending TCP connections
bind_ip = *0.0.0.0*	IP Address for server to bind to
bind_port = *6000*	Port for server to bind to
bind_timeout = *30*	Seconds to attempt bind before giving up
client_timeout = *60*	Timeout to read one chunk from a client external services
conn_timeout = *0.5*	Connection timeout to external services
devices = */srv/node*	Parent directory of where devices are mounted
disable_fallocate = *false*	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
disk_chunk_size = *65536*	Size of chunks to read/write to disk
eventlet_debug = *false*	If true, turn on debug logging for eventlet
expiring_objects_account_name = *expiring_objects*	No help text available for this option.
expiring_objects_container_divisor = *86400*	No help text available for this option.
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`.
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_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_max_line_length = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
log_name = *swift*	Label used when logging
log_statsd_default_sample_rate = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_host = *localhost*	If not set, the StatsD feature is disabled.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
log_statsd_port = *8125*	Port value for the StatsD server.
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.
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.
mount_check = *true*	Whether or not check if the devices are mounted to prevent accidentally writing to the root device

Configuration option = Default value	Description
network_chunk_size = *65536*	Size of chunks to read/write over the network
node_timeout = *3*	Request timeout to external services
swift_dir = */etc/swift*	Swift configuration directory
user = *swift*	User to run as
workers = *auto*	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.

Table 10.3. Description of configuration options for [app-object-server] in object-server.conf

Configuration option = Default value	Description
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
keep_cache_private = *false*	Allow non-public objects to stay in kernel's buffer cache
keep_cache_size = *5242880*	Largest object size to keep in buffer cache
max_upload_time = *86400*	Maximum time allowed to upload an object
mb_per_sync = *512*	On PUT requests, sync file every n MB
replication_concurrency = *4*	Set to restrict the number of concurrent incoming REPLICATION requests; set to 0 for unlimited
replication_failure_ratio = *1.0*	If the value of failures / successes of REPLICATION subrequests exceeds this ratio, the overall REPLICATION request will be aborted
replication_failure_threshold = *100*	The number of subrequest failures before the replication_failure_ratio is checked
replication_lock_timeout = *15*	Number of seconds to wait for an existing replication device lock before giving up.
replication_one_per_device = *True*	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_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.
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_level = *INFO*	Log level
set log_name = *object-server*	Label to use when logging
set log_requests = *true*	Whether or not to log requests
slow = *0*	If > 0, Minimum time in seconds for a PUT or DELETE request to complete
splice = *no*	No help text available for this option.
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.

Configuration option = Default value	Description
use = *egg:swift#object*	Entry point of paste.deploy in the server

Table 10.4. Description of configuration options for [pipeline-main] in object-server.conf

Configuration option = Default value	Description
pipeline = *healthcheck recon object-server*	No help text available for this option.

Table 10.5. Description of configuration options for [object-replicator] in object-server.conf

Configuration option = Default value	Description
concurrency = *1*	Number of replication workers to spawn
daemonize = *on*	Whether or not to run replication as a daemon
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).
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.
http_timeout = *60*	Maximum duration for an HTTP request
lockup_timeout = *1800*	Attempts to kill all workers if nothing replications for lockup_timeout seconds
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *object-replicator*	Label used when logging
node_timeout = *<whatever's in the DEFAULT section or 10>*	Request timeout to external services
reclaim_age = *604800*	Time elapsed in seconds before an object can be reclaimed
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
ring_check_interval = *15*	How often (in seconds) to check the ring
rsync_bwlimit = *0*	No help text available for this option.
rsync_error_log_line_length = *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
rsync_timeout = *900*	Max duration (seconds) of a partition rsync
run_pause = *30*	Time in seconds to wait between replication passes
stats_interval = *300*	Interval in seconds between logging replication statistics
sync_method = *rsync*	No help text available for this option.
vm_test_mode = *no*	Indicates that you are using a VM environment

Table 10.6. Description of configuration options for [object-updater] in object-server.conf

Configuration option = Default value	Description
concurrency = *1*	Number of replication workers to spawn

Configuration option = Default value	Description
interval = *300*	Minimum time for a pass to take
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *object-updater*	Label used when logging
node_timeout = *<whatever's in the DEFAULT section or 10>*	Request timeout to external services
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
slowdown = *0.01*	Time in seconds to wait between objects

Table 10.7. Description of configuration options for [object-auditor] in object-server.conf

Configuration option = Default value	Description
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`.
concurrency = *1*	Number of replication workers to spawn
disk_chunk_size = *65536*	Size of chunks to read/write to disk
files_per_second = *20*	Maximum files audited per second. Should be tuned according to individual system specs. 0 is unlimited.
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *object-auditor*	Label used when logging
log_time = *3600*	Frequency of status logs in seconds.
object_size_stats =	No help text available for this option.
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
zero_byte_files_per_second = *50*	Maximum zero byte files audited per second.

Table 10.8. Description of configuration options for [filter-healthcheck] in object-server.conf

Configuration option = Default value	Description
disable_path =	No help text available for this option.
use = *egg:swift#healthcheck*	Entry point of paste.deploy in the server

Table 10.9. Description of configuration options for [filter-recon] in object-server.conf

Configuration option = Default value	Description
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.
use = *egg:swift#recon*	Entry point of paste.deploy in the server

Table 10.10. Description of configuration options for [filter-xprofile] in object-server.conf

Configuration option = Default value	Description
dump_interval = *5.0*	No help text available for this option.
dump_timestamp = *false*	No help text available for this option.
flush_at_shutdown = *false*	No help text available for this option.
log_filename_prefix = */tmp/log/swift/profile/default.profile*	No help text available for this option.
path = */__profile__*	No help text available for this option.
profile_module = *eventlet.green.profile*	No help text available for this option.
unwind = *false*	No help text available for this option.
use = *egg:swift#xprofile*	Entry point of paste.deploy in the server

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
# devices = /srv/node
# mount_check = true
# disable_fallocate = false
# expiring_objects_container_divisor = 86400
# expiring_objects_account_name = expiring_objects
#
# 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
# The following caps the length of log lines to the value given; no limit if
# set to 0, the default.
# log_max_line_length = 0
#
# 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 =
#
# 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
#
# Time to wait while attempting to connect to another backend node.
# conn_timeout = 0.5
# Time to wait while sending each chunk of data to another backend node.
# node_timeout = 3
# Time to wait while receiving each chunk of data from a client or another
# backend node.
# client_timeout = 60
#
# network_chunk_size = 65536
# disk_chunk_size = 65536

[pipeline:main]
pipeline = healthcheck recon object-server

[app:object-server]
use = egg:swift#object
# You can override the default log routing for this app here:
# set log_name = object-server
# set log_facility = LOG_LOCAL0
# 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 = 5242880
#
# 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
#
# Use splice() for zero-copy object GETs. This requires Linux kernel
# version 3.0 or greater. If you set "splice = yes" but the kernel
# does not support it, error messages will appear in the object server
# logs at startup, but your object servers should continue to function.
#
# splice = no

[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
#
# handoffs_first and handoff_delete are options for a special case
# such as disk full in the cluster. These two options SHOULD NOT BE
# CHANGED, except for such an extreme situations. (e.g. disks filled up
# or are about to fill up. Anyway, DO NOT let your drives fill up)
# handoffs_first is the flag to replicate handoffs prior to canonical
# partitions. It allows to force syncing and deleting handoffs quickly.
# If set to a True value(e.g. "True" or "1"), partitions
# that are not supposed to be on the node will be replicated first.
# handoffs_first = False
#
# handoff_delete is the number of replicas which are ensured in swift.
# If the number less than the number of replicas is set, object-replicator
# could delete local handoffs even if all replicas are not ensured in the
```

```
# cluster. Object-replicator would remove local handoff partition directories
# after syncing partition when the number of successful responses is greater
# than or equal to this number. By default(auto), handoff partitions will be
# removed  when it has successfully replicated to all the canonical nodes.
# handoff_delete = auto

[object-reconstructor]
# You can override the default log routing for this app here (don't use set!):
# Unless otherwise noted, each setting below has the same meaning as described
# in the [object-replicator] section, however these settings apply to the EC
# reconstructor
#
# log_name = object-reconstructor
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# daemonize = on
# run_pause = 30
# concurrency = 1
# stats_interval = 300
# node_timeout = 10
# http_timeout = 60
# lockup_timeout = 1800
# reclaim_age = 604800
# ring_check_interval = 15
# recon_cache_path = /var/cache/swift
# handoffs_first = False

[object-updater]
# You can override the default log routing for this app here (don't use set!):
# log_name = object-updater
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# interval = 300
# concurrency = 1
# node_timeout = <whatever's in the DEFAULT section or 10>
# slowdown will sleep that amount between objects
# slowdown = 0.01
#
# recon_cache_path = /var/cache/swift

[object-auditor]
# You can override the default log routing for this app here (don't use set!):
# log_name = object-auditor
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# You can set the disk chunk size that the auditor uses making it larger if
# you like for more efficient local auditing of larger objects
# disk_chunk_size = 65536
# files_per_second = 20
# concurrency = 1
# bytes_per_second = 10000000
# log_time = 3600
# zero_byte_files_per_second = 50
# recon_cache_path = /var/cache/swift
```

```
# Takes a comma separated list of ints. If set, the object auditor will
# increment a counter for every object whose size is <= to the given break
# points and report the result after a full scan.
# object_size_stats =

# Note: Put it at the beginning of the pipleline to profile all middleware.
 But
# it is safer to put this after healthcheck.
[filter:xprofile]
use = egg:swift#xprofile
# This option enable you to switch profilers which should inherit from python
# standard profiler. Currently the supported value can be 'cProfile',
# 'eventlet.green.profile' etc.
# profile_module = eventlet.green.profile
#
# This prefix will be used to combine process ID and timestamp to name the
# profile data file.  Make sure the executing user has permission to write
# into this path (missing path segments will be created, if necessary).
# If you enable profiling in more than one type of daemon, you must override
# it with an unique value like: /var/log/swift/profile/object.profile
# log_filename_prefix = /tmp/log/swift/profile/default.profile
#
# the profile data will be dumped to local disk based on above naming rule
# in this interval.
# dump_interval = 5.0
#
# Be careful, this option will enable profiler to dump data into the file with
# time stamp which means there will be lots of files piled up in the
 directory.
# dump_timestamp = false
#
# This is the path of the URL to access the mini web UI.
# path = /__profile__
#
# Clear the data when the wsgi server shutdown.
# flush_at_shutdown = false
#
# unwind the iterator of applications
# unwind = false
```

Object expirer configuration

Find an example object expirer configuration at `etc/object-expirer.conf-sample` in the source code repository.

The available configuration options are:

Table 10.11. Description of configuration options for [DEFAULT] in object-expirer.conf

Configuration option = Default value	Description
`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_facility` = *LOG_LOCAL0*	Syslog log facility
`log_level` = *INFO*	Logging level

Configuration option = Default value	Description
log_max_line_length = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
log_name = *swift*	Label used when logging
log_statsd_default_sample_rate = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_host = *localhost*	If not set, the StatsD feature is disabled.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
log_statsd_port = *8125*	Port value for the StatsD server.
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.
swift_dir = */etc/swift*	Swift configuration directory
user = *swift*	User to run as

Table 10.12. Description of configuration options for [app-proxy-server] in object-expirer.conf

Configuration option = Default value	Description
use = *egg:swift#proxy*	Entry point of paste.deploy in the server

Table 10.13. Description of configuration options for [filter-cache] in object-expirer.conf

Configuration option = Default value	Description
use = *egg:swift#memcache*	Entry point of paste.deploy in the server

Table 10.14. Description of configuration options for [filter-catch_errors] in object-expirer.conf

Configuration option = Default value	Description
use = *egg:swift#catch_errors*	Entry point of paste.deploy in the server

Table 10.15. Description of configuration options for [filter-proxy-logging] in object-expirer.conf

Configuration option = Default value	Description
access_log_address = */dev/log*	No help text available for this option.
access_log_facility = *LOG_LOCAL0*	No help text available for this option.
access_log_headers = *false*	No help text available for this option.
access_log_headers_only =	If access_log_headers is True and access_log_headers_only is set only these headers are logged. Multiple headers can be defined as comma separated list like this: access_log_headers_only = Host, X-Object-Meta-Mtime
access_log_level = *INFO*	No help text available for this option.
access_log_name = *swift*	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_host = *localhost*	No help text available for this option.
access_log_statsd_metric_prefix =	No help text available for this option.

Configuration option = Default value	Description
access_log_statsd_port = *8125*	No help text available for this option.
access_log_statsd_sample_rate_factor = *1.0*	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.
log_statsd_valid_http_methods = *GET,HEAD,POST,PUT,DELETE,COPY,OPTIONS*	No help text available for this option.
logged with access_log_headers = *True.*	No help text available for this option.
reveal_sensitive_prefix = *16*	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.
use = *egg:swift#proxy_logging*	Entry point of paste.deploy in the server

Table 10.16. Description of configuration options for [object-expirer] in object-expirer.conf

Configuration option = Default value	Description
auto_create_account_prefix = *.*	Prefix to use when automatically creating accounts
concurrency = *1*	Number of replication workers to spawn
expiring_objects_account_name = *expiring_objects*	No help text available for this option.
interval = *300*	Minimum time for a pass to take
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.
processes = *0*	No help text available for this option.
reclaim_age = *604800*	Time elapsed in seconds before an object can be reclaimed
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
report_interval = *300*	No help text available for this option.

Table 10.17. Description of configuration options for [pipeline-main] in object-expirer.conf

Configuration option = Default value	Description
pipeline = *catch_errors proxy-logging 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
# The following caps the length of log lines to the value given; no limit if
# set to 0, the default.
# log_max_line_length = 0
```

```
#
# 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 overide 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
# The expirer will re-attempt expiring if the source object is not available
# up to reclaim_age seconds before it gives up and deletes the entry in the
# queue.
# reclaim_age = 604800
# recon_cache_path = /var/cache/swift

[pipeline:main]
pipeline = catch_errors proxy-logging 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

[filter:proxy-logging]
```

```
use = egg:swift#proxy_logging
# If not set, logging directives from [DEFAULT] without "access_" will be used
# access_log_name = swift
# access_log_facility = LOG_LOCAL0
# access_log_level = INFO
# access_log_address = /dev/log
#
# If set, access_log_udp_host will override access_log_address
# access_log_udp_host =
# access_log_udp_port = 514
#
# You can use log_statsd_* from [DEFAULT] or override them here:
# access_log_statsd_host = localhost
# access_log_statsd_port = 8125
# access_log_statsd_default_sample_rate = 1.0
# access_log_statsd_sample_rate_factor = 1.0
# access_log_statsd_metric_prefix =
# access_log_headers = false
#
# If access_log_headers is True and access_log_headers_only is set only
# these headers are logged. Multiple headers can be defined as comma separated
# list like this: access_log_headers_only = Host, X-Object-Meta-Mtime
# access_log_headers_only =
#
# By default, the X-Auth-Token is logged. To obscure the value,
# set reveal_sensitive_prefix to the number of characters to log.
# For example, if set to 12, only the first 12 characters of the
# token appear in the log. An unauthorized access of the log file
# won't allow unauthorized usage of the token. However, the first
# 12 or so characters is unique enough that you can trace/debug
# token usage. Set to 0 to suppress the token completely (replaced
# by '...' in the log).
# Note: reveal_sensitive_prefix will not affect the value
# logged with access_log_headers=True.
# reveal_sensitive_prefix = 16
#
# What HTTP methods are allowed for StatsD logging (comma-sep); request
 methods
# not in this list will have "BAD_METHOD" for the <verb> portion of the
 metric.
# log_statsd_valid_http_methods = GET,HEAD,POST,PUT,DELETE,COPY,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 10.18. Description of configuration options for [DEFAULT] in container-server.conf

Configuration option = Default value	Description
`allowed_sync_hosts` = *127.0.0.1*	No help text available for this option.
`backlog` = *4096*	Maximum number of allowed pending TCP connections
`bind_ip` = *0.0.0.0*	IP Address for server to bind to
`bind_port` = *6001*	Port for server to bind to

Configuration option = Default value	Description
bind_timeout = *30*	Seconds to attempt bind before giving up
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
devices = */srv/node*	Parent directory of where devices are mounted
disable_fallocate = *false*	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
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`.
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_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_max_line_length = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
log_name = *swift*	Label used when logging
log_statsd_default_sample_rate = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_host = *localhost*	If not set, the StatsD feature is disabled.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
log_statsd_port = *8125*	Port value for the StatsD server.
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.
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.
mount_check = *true*	Whether or not check if the devices are mounted to prevent accidentally writing to the root device
swift_dir = */etc/swift*	Swift configuration directory
user = *swift*	User to run as

Configuration option = Default value	Description
workers = *auto*	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.

Table 10.19. Description of configuration options for [app-container-server] in container-server.conf

Configuration option = Default value	Description
allow_versions = *false*	Enable/Disable object versioning feature
auto_create_account_prefix = *.*	Prefix to use when automatically creating accounts
conn_timeout = *0.5*	Connection timeout to external services
node_timeout = *3*	Request timeout to external services
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.
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_level = *INFO*	Log level
set log_name = *container-server*	Label to use when logging
set log_requests = *true*	Whether or not to log requests
use = *egg:swift#container*	Entry point of paste.deploy in the server

Table 10.20. Description of configuration options for [pipeline-main] in container-server.conf

Configuration option = Default value	Description
pipeline = *healthcheck recon container-server*	No help text available for this option.

Table 10.21. Description of configuration options for [container-replicator] in container-server.conf

Configuration option = Default value	Description
concurrency = *8*	Number of replication workers to spawn
conn_timeout = *0.5*	Connection timeout to external services
interval = *30*	Minimum time for a pass to take
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *container-replicator*	Label used when logging
max_diffs = *100*	Caps how long the replicator spends trying to sync a database per pass
node_timeout = *10*	Request timeout to external services
per_diff = *1000*	Limit number of items to get per diff
reclaim_age = *604800*	Time elapsed in seconds before an object can be reclaimed
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
run_pause = *30*	Time in seconds to wait between replication passes
vm_test_mode = *no*	Indicates that you are using a VM environment

Table 10.22. Description of configuration options for [container-updater] in container-server.conf

Configuration option = Default value	Description
account_suppression_time = 60	Seconds to suppress updating an account that has generated an error (timeout, not yet found, etc.)
concurrency = 4	Number of replication workers to spawn
conn_timeout = 0.5	Connection timeout to external services
interval = 300	Minimum time for a pass to take
log_address = /dev/log	Location where syslog sends the logs to
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_name = container-updater	Label used when logging
node_timeout = 3	Request timeout to external services
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored
slowdown = 0.01	Time in seconds to wait between objects

Table 10.23. Description of configuration options for [container-auditor] in container-server.conf

Configuration option = Default value	Description
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.
interval = 1800	Minimum time for a pass to take
log_address = /dev/log	Location where syslog sends the logs to
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_name = container-auditor	Label used when logging
recon_cache_path = /var/cache/swift	Directory where stats for a few items will be stored

Table 10.24. Description of configuration options for [container-sync] in container-server.conf

Configuration option = Default value	Description
conn_timeout = 5	Connection timeout to external services
container_time = 60	Maximum amount of time to spend syncing each container
internal_client_conf_path = /etc/swift/internal-client.conf	No help text available for this option.
interval = 300	Minimum time for a pass to take
log_address = /dev/log	Location where syslog sends the logs to
log_facility = LOG_LOCAL0	Syslog log facility
log_level = INFO	Logging level
log_name = container-sync	Label used when logging

Configuration option = Default value	Description
request_tries = *3*	No help text available for this option.
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.

Table 10.25. Description of configuration options for [filter-healthcheck] in container-server.conf

Configuration option = Default value	Description
disable_path =	No help text available for this option.
use = *egg:swift#healthcheck*	Entry point of paste.deploy in the server

Table 10.26. Description of configuration options for [filter-recon] in container-server.conf

Configuration option = Default value	Description
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
use = *egg:swift#recon*	Entry point of paste.deploy in the server

Table 10.27. Description of configuration options for [filter-xprofile] in container-server.conf

Configuration option = Default value	Description
dump_interval = *5.0*	No help text available for this option.
dump_timestamp = *false*	No help text available for this option.
flush_at_shutdown = *false*	No help text available for this option.
log_filename_prefix = */tmp/log/swift/profile/default.profile*	No help text available for this option.
path = */__profile__*	No help text available for this option.
profile_module = *eventlet.green.profile*	No help text available for this option.
unwind = *false*	No help text available for this option.
use = *egg:swift#xprofile*	Entry point of paste.deploy in the server

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
# The following caps the length of log lines to the value given; no limit if
# set to 0, the default.
# log_max_line_length = 0
#
# 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
# Note: if the parameter 'interval' is defined then it will be used in place
# of run_pause.
# 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
#
# Maximum amount of time in seconds for the connection attempt
# conn_timeout = 5
# Server errors from requests will be retried by default
# request_tries = 3
#
# Internal client config file path
# internal_client_conf_path = /etc/swift/internal-client.conf

# Note: Put it at the beginning of the pipeline to profile all middleware. But
# it is safer to put this after healthcheck.
[filter:xprofile]
use = egg:swift#xprofile
# This option enable you to switch profilers which should inherit from python
# standard profiler. Currently the supported value can be 'cProfile',
# 'eventlet.green.profile' etc.
# profile_module = eventlet.green.profile
#
# This prefix will be used to combine process ID and timestamp to name the
# profile data file.  Make sure the executing user has permission to write
# into this path (missing path segments will be created, if necessary).
# If you enable profiling in more than one type of daemon, you must override
# it with an unique value like: /var/log/swift/profile/container.profile
# log_filename_prefix = /tmp/log/swift/profile/default.profile
#
# the profile data will be dumped to local disk based on above naming rule
# in this interval.
# dump_interval = 5.0
```

```
#
# Be careful, this option will enable profiler to dump data into the file with
# time stamp which means there will be lots of files piled up in the
 directory.
# dump_timestamp = false
#
# This is the path of the URL to access the mini web UI.
# path = /__profile__
#
# Clear the data when the wsgi server shutdown.
# flush_at_shutdown = false
#
# unwind the iterator of applications
# unwind = false
```

Container sync realms configuration

Find an example container sync realms configuration at `etc/container-sync-realms.conf-sample` in the source code repository.

The available configuration options are:

Table 10.28. Description of configuration options for [DEFAULT] in container-sync-realms.conf

Configuration option = Default value	Description
`mtime_check_interval` = *300*	No help text available for this option.

Table 10.29. Description of configuration options for [realm1] in container-sync-realms.conf

Configuration option = Default value	Description
`cluster_name1` = *https://host1/v1/*	No help text available for this option.
`cluster_name2` = *https://host2/v1/*	No help text available for this option.
`key` = *realm1key*	No help text available for this option.
`key2` = *realm1key2*	No help text available for this option.

Table 10.30. Description of configuration options for [realm2] in container-sync-realms.conf

Configuration option = Default value	Description
`cluster_name3` = *https://host3/v1/*	No help text available for this option.
`cluster_name4` = *https://host4/v1/*	No help text available for this option.
`key` = *realm2key*	No help text available for this option.
`key2` = *realm2key2*	No help text available for this option.

Sample container sync realms configuration file

```
# [DEFAULT]
# The number of seconds between checking the modified time of this config file
# for changes and therefore reloading it.
# mtime_check_interval = 300
```

```
# [realm1]
# key = realm1key
# key2 = realm1key2
# cluster_name1 = https://host1/v1/
# cluster_name2 = https://host2/v1/
#
# [realm2]
# key = realm2key
# key2 = realm2key2
# cluster_name3 = https://host3/v1/
# cluster_name4 = https://host4/v1/


# Each section name is the name of a sync realm. A sync realm is a set of
# clusters that have agreed to allow container syncing with each other. Realm
# names will be considered case insensitive.
#
# The key is the overall cluster-to-cluster key used in combination with the
# external users' key that they set on their containers' X-Container-Sync-Key
# metadata header values. These keys will be used to sign each request the
# container sync daemon makes and used to validate each incoming container
 sync
# request.
#
# The key2 is optional and is an additional key incoming requests will be
# checked against. This is so you can rotate keys if you wish; you move the
# existing key to key2 and make a new key value.
#
# Any values in the realm section whose names begin with cluster_ will
 indicate
# the name and endpoint of a cluster and will be used by external users in
# their containers' X-Container-Sync-To metadata header values with the format
# "realm_name/cluster_name/container_name". Realm and cluster names are
# considered case insensitive.
#
# The endpoint is what the container sync daemon will use when sending out
# requests to that cluster. Keep in mind this endpoint must be reachable by
 all
# container servers, since that is where the container sync daemon runs. Note
# the the endpoint ends with /v1/ and that the container sync daemon will then
# add the account/container/obj name after that.
#
# Distribute this container-sync-realms.conf file to all your proxy servers
# and container servers.
```

Container reconciler configuration

Find an example container sync realms configuration at `etc/container-reconciler.conf-sample` in the source code repository.

The available configuration options are:

Table 10.31. Description of configuration options for [DEFAULT] in container-reconciler.conf

Configuration option = Default value	Description
`log_address` = */dev/log*	Location where syslog sends the logs to

Configuration option = Default value	Description
log_custom_handlers =	Comma-separated list of functions to call to setup custom log handlers.
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *swift*	Label used when logging
log_statsd_default_sample_rate = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_host = *localhost*	If not set, the StatsD feature is disabled.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
log_statsd_port = *8125*	Port value for the StatsD server.
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.
swift_dir = */etc/swift*	Swift configuration directory
user = *swift*	User to run as

Table 10.32. Description of configuration options for [app-proxy-server] in container-reconciler.conf

Configuration option = Default value	Description
use = *egg:swift#proxy*	Entry point of paste.deploy in the server

Table 10.33. Description of configuration options for [container-reconciler] in container-reconciler.conf

Configuration option = Default value	Description
interval = *30*	Minimum time for a pass to take
reclaim_age = *604800*	Time elapsed in seconds before an object can be reclaimed
request_tries = *3*	No help text available for this option.

Table 10.34. Description of configuration options for [filter-cache] in container-reconciler.conf

Configuration option = Default value	Description
use = *egg:swift#memcache*	Entry point of paste.deploy in the server

Table 10.35. Description of configuration options for [filter-catch_errors] in container-reconciler.conf

Configuration option = Default value	Description
use = *egg:swift#catch_errors*	Entry point of paste.deploy in the server

Table 10.36. Description of configuration options for [filter-proxy-logging] in container-reconciler.conf

Configuration option = Default value	Description
use = *egg:swift#proxy_logging*	Entry point of paste.deploy in the server

Table 10.37. Description of configuration options for [pipeline-main] in container-reconciler.conf

Configuration option = Default value	Description
pipeline = *catch_errors proxy-logging cache proxy-server*	No help text available for this option.

Sample container sync reconciler 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 =

[container-reconciler]
# The reconciler will re-attempt reconciliation if the source object is not
# available up to reclaim_age seconds before it gives up and deletes the entry
# in the queue.
# reclaim_age = 604800
# The cycle time of the daemon
# interval = 30
# Server errors from requests will be retried by default
# request_tries = 3

[pipeline:main]
pipeline = catch_errors proxy-logging 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:proxy-logging]
use = egg:swift#proxy_logging

[filter:catch_errors]
```

```
use = egg:swift#catch_errors
# See proxy-server.conf-sample for options
```

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 10.38. Description of configuration options for [DEFAULT] in account-server.conf

Configuration option = Default value	Description
`backlog` = *4096*	Maximum number of allowed pending TCP connections
`bind_ip` = *0.0.0.0*	IP Address for server to bind to
`bind_port` = *6002*	Port for server to bind to
`bind_timeout` = *30*	Seconds to attempt bind before giving up
`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
`devices` = */srv/node*	Parent directory of where devices are mounted
`disable_fallocate` = *false*	Disable "fast fail" fallocate checks if the underlying filesystem does not support it.
`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`.
`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_facility` = *LOG_LOCAL0*	Syslog log facility
`log_level` = *INFO*	Logging level
`log_max_line_length` = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
`log_name` = *swift*	Label used when logging
`log_statsd_default_sample_rate` = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
`log_statsd_host` = *localhost*	If not set, the StatsD feature is disabled.
`log_statsd_metric_prefix` =	Value will be prepended to every metric sent to the StatsD server.
`log_statsd_port` = *8125*	Port value for the StatsD server.

Configuration option = Default value	Description
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.
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.
mount_check = *true*	Whether or not check if the devices are mounted to prevent accidentally writing to the root device
swift_dir = */etc/swift*	Swift configuration directory
user = *swift*	User to run as
workers = *auto*	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.

Table 10.39. Description of configuration options for [app-account-server] in account-server.conf

Configuration option = Default value	Description
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.
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_level = *INFO*	Log level
set log_name = *account-server*	Label to use when logging
set log_requests = *true*	Whether or not to log requests
use = *egg:swift#account*	Entry point of paste.deploy in the server

Table 10.40. Description of configuration options for [pipeline-main] in account-server.conf

Configuration option = Default value	Description
pipeline = *healthcheck recon account-server*	No help text available for this option.

Table 10.41. Description of configuration options for [account-replicator] in account-server.conf

Configuration option = Default value	Description
concurrency = *8*	Number of replication workers to spawn
conn_timeout = *0.5*	Connection timeout to external services
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

Configuration option = Default value	Description
interval = *30*	Minimum time for a pass to take
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *account-replicator*	Label used when logging
max_diffs = *100*	Caps how long the replicator spends trying to sync a database per pass
node_timeout = *10*	Request timeout to external services
per_diff = *1000*	Limit number of items to get per diff
reclaim_age = *604800*	Time elapsed in seconds before an object can be reclaimed
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
run_pause = *30*	Time in seconds to wait between replication passes
vm_test_mode = *no*	Indicates that you are using a VM environment

Table 10.42. Description of configuration options for [account-auditor] in account-server.conf

Configuration option = Default value	Description
accounts_per_second = *200*	Maximum accounts audited per second. Should be tuned according to individual system specs. 0 is unlimited.
interval = *1800*	Minimum time for a pass to take
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *account-auditor*	Label used when logging
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored

Table 10.43. Description of configuration options for [account-reaper] in account-server.conf

Configuration option = Default value	Description
concurrency = *25*	Number of replication workers to spawn
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.
interval = *3600*	Minimum time for a pass to take
log_address = */dev/log*	Location where syslog sends the logs to
log_facility = *LOG_LOCAL0*	Syslog log facility
log_level = *INFO*	Logging level
log_name = *account-reaper*	Label used when logging
node_timeout = *10*	Request timeout to external services
reap_warn_after = *2592000*	No help text available for this option.

Table 10.44. Description of configuration options for [filter-healthcheck] in account-server.conf

Configuration option = Default value	Description
disable_path =	No help text available for this option.
use = *egg:swift#healthcheck*	Entry point of paste.deploy in the server

Table 10.45. Description of configuration options for [filter-recon] in account-server.conf

Configuration option = Default value	Description
recon_cache_path = */var/cache/swift*	Directory where stats for a few items will be stored
use = *egg:swift#recon*	Entry point of paste.deploy in the server

Table 10.46. Description of configuration options for [filter-xprofile] in account-server.conf

Configuration option = Default value	Description
dump_interval = *5.0*	No help text available for this option.
dump_timestamp = *false*	No help text available for this option.
flush_at_shutdown = *false*	No help text available for this option.
log_filename_prefix = */tmp/log/swift/profile/default.profile*	No help text available for this option.
path = */__profile__*	No help text available for this option.
profile_module = *eventlet.green.profile*	No help text available for this option.
unwind = *false*	No help text available for this option.
use = *egg:swift#xprofile*	Entry point of paste.deploy in the server

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
# The following caps the length of log lines to the value given; no limit if
# set to 0, the default.
```

```
# log_max_line_length = 0
#
# 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
# Note: if the parameter 'interval' is defined then it will be used in place
# of run_pause.
# 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
# recon_cache_path = /var/cache/swift

[account-reaper]
# You can override the default log routing for this app here (don't use set!):
# log_name = account-reaper
# log_facility = LOG_LOCAL0
# log_level = INFO
# log_address = /dev/log
#
# concurrency = 25
# interval = 3600
# node_timeout = 10
# conn_timeout = 0.5
```

```
#
# 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.
# delay_reaping = 0
#
# If the account fails to be be reaped due to a persistent error, the
# account reaper will log a message such as:
#     Account <name> has not been reaped since <date>
# You can search logs for this message if space is not being reclaimed
# after you delete account(s).
# Default is 2592000 seconds (30 days). This is in addition to any time
# requested by delay_reaping.
# reap_warn_after = 2592000

# Note: Put it at the beginning of the pipeline to profile all middleware. But
# it is safer to put this after healthcheck.
[filter:xprofile]
use = egg:swift#xprofile
# This option enable you to switch profilers which should inherit from python
# standard profiler. Currently the supported value can be 'cProfile',
# 'eventlet.green.profile' etc.
# profile_module = eventlet.green.profile
#
# This prefix will be used to combine process ID and timestamp to name the
# profile data file.  Make sure the executing user has permission to write
# into this path (missing path segments will be created, if necessary).
# If you enable profiling in more than one type of daemon, you must override
# it with an unique value like: /var/log/swift/profile/account.profile
# log_filename_prefix = /tmp/log/swift/profile/default.profile
#
# the profile data will be dumped to local disk based on above naming rule
# in this interval.
# dump_interval = 5.0
#
# Be careful, this option will enable profiler to dump data into the file with
# time stamp which means there will be lots of files piled up in the
 directory.
# dump_timestamp = false
#
# This is the path of the URL to access the mini web UI.
# path = /__profile__
#
# Clear the data when the wsgi server shutdown.
# flush_at_shutdown = false
#
# unwind the iterator of applications
# unwind = false
```

Proxy server configuration

Find an example proxy server configuration at `etc/proxy-server.conf-sample` in the source code repository.

The available configuration options are:

Table 10.47. Description of configuration options for [DEFAULT] in proxy-server.conf

Configuration option = Default value	Description
admin_key = *secret_admin_key*	to use for admin calls that are HMAC signed. Default is empty, which will disable admin calls to /info. the proxy server. For most cases, this should be `egg:swift#proxy`. request whenever it has to failover to a handoff node
backlog = *4096*	Maximum number of allowed pending TCP connections
bind_ip = *0.0.0.0*	IP Address for server to bind to
bind_port = *8080*	Port for server to bind to
bind_timeout = *30*	Seconds to attempt bind before giving up
cert_file = */etc/swift/proxy.crt*	to the ssl .crt. This should be enabled for testing purposes only.
client_timeout = *60*	Timeout to read one chunk from a client external services
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
disallowed_sections = *swift.valid_api_versions, container_quotas, tempurl*	No help text available for this option.
eventlet_debug = *false*	If true, turn on debug logging for eventlet
expiring_objects_account_name = *expiring_objects*	No help text available for this option.
expiring_objects_container_divisor = *86400*	No help text available for this option.
expose_info = *true*	Enables exposing configuration settings via HTTP GET /info.
key_file = */etc/swift/proxy.key*	to the ssl .key. This should be enabled for testing purposes only.
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_facility = *LOG_LOCAL0*	Syslog log facility
log_headers = *false*	No help text available for this option.
log_level = *INFO*	Logging level
log_max_line_length = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
log_name = *swift*	Label used when logging
log_statsd_default_sample_rate = *1.0*	Defines the probability of sending a sample for any given event or timing measurement.
log_statsd_host = *localhost*	If not set, the StatsD feature is disabled.
log_statsd_metric_prefix =	Value will be prepended to every metric sent to the StatsD server.
log_statsd_port = *8125*	Port value for the StatsD server.
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_udp_host =	If not set, the UDP receiver for syslog is disabled.
log_udp_port = *514*	Port value for UDP receiver, if enabled.

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.
strict_cors_mode = *True*	No help text available for this option.
swift_dir = */etc/swift*	Swift configuration directory
trans_id_suffix =	No help text available for this option.
user = *swift*	User to run as
workers = *auto*	a much higher value, one can reduce the impact of slow file system operations in one request from negatively impacting other requests.

Table 10.48. Description of configuration options for [app-proxy-server] in proxy-server.conf

Configuration option = Default value	Description
account_autocreate = *false*	If set to 'true' authorized accounts that do not yet exist within the Swift cluster will be automatically created.
allow_account_management = *false*	Whether account PUTs and DELETEs are even callable
auto_create_account_prefix = *.*	Prefix to use when automatically creating accounts
client_chunk_size = *65536*	Chunk size to read from clients
conn_timeout = *0.5*	Connection timeout to external services
deny_host_headers =	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
log_handoffs = *true*	No help text available for this option.
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.
max_large_object_get_time = *86400*	No help text available for this option.
node_timeout = *10*	Request timeout to external services
object_chunk_size = *65536*	Chunk size to read from object servers
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.
post_quorum_timeout = *0.5*	No help text available for this option.
put_queue_depth = *10*	No help text available for this option.
read_affinity = *r1z1=100, r1z2=200, r2=300*	No help text available for this option.
recheck_account_existence = *60*	Cache timeout in seconds to send memcached for account existence

Configuration option = Default value	Description
recheck_container_existence = 60	Cache timeout in seconds to send memcached for container existence
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
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
set log_address = /dev/log	Location where syslog sends the logs to
set log_facility = LOG_LOCAL0	Syslog log facility
set log_level = INFO	Log level
set log_name = proxy-server	Label to use when logging
sorting_method = shuffle	No help text available for this option.
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-container-meta-temp-url-key, x-container-meta-temp-url-key-2, x-account-access-control	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
timing_expiry = 300	No help text available for this option.
use = egg:swift#proxy	Entry point of paste.deploy in the server
write_affinity = r1, r2	This setting lets you trade data distribution for throughput. It makes the proxy server prefer local back-end servers for object PUT requests over non-local ones. Note that only object PUT requests are affected by the write_affinity setting; POST, GET, HEAD, DELETE, OPTIONS, and account/container PUT requests are not affected. The 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 the write_affinity_node_count setting) 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
write_affinity_node_count = 2 * replicas	This setting is only useful in conjunction with write_affinity; it governs how many local object servers will be tried before falling back to 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

Table 10.49. Description of configuration options for [pipeline-main] in proxy-server.conf

Configuration option = Default value	Description
pipeline = catch_errors gatekeeper healthcheck proxy-logging cache container_sync bulk tempurl ratelimit tempauth container-quotas account-quotas slo dlo proxy-logging proxy-server	No help text available for this option.

Table 10.50. Description of configuration options for [filter-account-quotas] in proxy-server.conf

Configuration option = Default value	Description
use = *egg:swift#account_quotas*	Entry point of paste.deploy in the server

Table 10.51. Description of configuration options for [filter-authtoken] in proxy-server.conf

Configuration option = Default value	Description
admin_password = *password*	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.
auth_uri = *http://keystonehost:5000/*	No help text available for this option.
cache = *swift.cache*	No help text available for this option.
delay_auth_decision = *False*	No help text available for this option.
identity_uri = *http://keystonehost:35357/*	No help text available for this option.
include_service_catalog = *False*	No help text available for this option.

Table 10.52. Description of configuration options for [filter-cache] in proxy-server.conf

Configuration option = Default value	Description
memcache_max_connections = *2*	Max number of connections to each memcached server per worker services
memcache_serialization_support = *2*	Sets how memcache values are serialized and deserialized
memcache_servers = *127.0.0.1:11211*	Comma-separated list of memcached servers ip:port services
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *cache*	Label to use when logging
use = *egg:swift#memcache*	Entry point of paste.deploy in the server

Table 10.53. Description of configuration options for [filter-catch_errors] in proxy-server.conf

Configuration option = Default value	Description
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *catch_errors*	Label to use when logging
use = *egg:swift#catch_errors*	Entry point of paste.deploy in the server

Table 10.54. Description of configuration options for [filter-container_sync] in proxy-server.conf

Configuration option = Default value	Description
allow_full_urls = *true*	No help text available for this option.
current = *//REALM/CLUSTER*	No help text available for this option.
use = *egg:swift#container_sync*	Entry point of paste.deploy in the server

Table 10.55. Description of configuration options for [filter-dlo] in proxy-server.conf

Configuration option = Default value	Description
max_get_time = *86400*	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 = *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
use = *egg:swift#dlo*	Entry point of paste.deploy in the server

Table 10.56. Description of configuration options for [filter-gatekeeper] in proxy-server.conf

Configuration option = Default value	Description
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *gatekeeper*	Label to use when logging
use = *egg:swift#gatekeeper*	Entry point of paste.deploy in the server

Table 10.57. Description of configuration options for [filter-healthcheck] in proxy-server.conf

Configuration option = Default value	Description
disable_path =	No help text available for this option.
use = *egg:swift#healthcheck*	Entry point of paste.deploy in the server

Table 10.58. Description of configuration options for [filter-keystoneauth] in proxy-server.conf

Configuration option = Default value	Description
allow_names_in_acls = *true*	The backwards compatible behavior can be disabled by setting this option to False.
allow_overrides = *true*	This option allows middleware higher in the WSGI pipeline to override auth processing, useful for middleware such as tempurl and formpost. If you know you are not going to use such middleware and you want a bit of extra security, you can set this to False.
default_domain_id = *default*	Name of the default domain. It is identified by its UUID, which by default has the value "default".

Configuration option = Default value	Description
is_admin = *false*	If this option is set to True, it allows to give a user whose username is the same as the project name and who has any role in the project access rights elevated to be the same as if the user had one of the operator_roles. Note that the condition compares names rather than UUIDs. This option is deprecated. It is False by default.
operator_roles = *admin, swiftoperator*	Operator role defines the user which is allowed to manage a tenant and create containers or give ACL to others. This parameter may be prefixed with an appropriate prefix.
reseller_admin_role = *ResellerAdmin*	The reseller admin role gives the ability to create and delete accounts.
reseller_prefix = *AUTH*	The naming scope for the auth service. Swift
service_roles =	When present, this option requires that the X-Service-Token header supplies a token from a user who has a role listed in service_roles. This parameter may be prefixed with an appropriate prefix.
use = *egg:swift#keystoneauth*	Entry point of paste.deploy in the server

Table 10.59. Description of configuration options for [filter-list-endpoints] in proxy-server.conf

Configuration option = Default value	Description
list_endpoints_path = */endpoints/*	No help text available for this option.
use = *egg:swift#list_endpoints*	Entry point of paste.deploy in the server

Table 10.60. Description of configuration options for [filter-proxy-logging] in proxy-server.conf

Configuration option = Default value	Description
access_log_address = */dev/log*	No help text available for this option.
access_log_facility = *LOG_LOCAL0*	No help text available for this option.
access_log_headers = *false*	No help text available for this option.
access_log_headers_only =	If access_log_headers is True and access_log_headers_only is set only these headers are logged. Multiple headers can be defined as comma separated list like this: access_log_headers_only = Host, X-Object-Meta-Mtime
access_log_level = *INFO*	No help text available for this option.
access_log_name = *swift*	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_host = *localhost*	No help text available for this option.
access_log_statsd_metric_prefix =	No help text available for this option.
access_log_statsd_port = *8125*	No help text available for this option.
access_log_statsd_sample_rate_factor = *1.0*	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.
log_statsd_valid_http_methods = *GET,HEAD,POST,PUT,DELETE,COPY,OPTIONS*	No help text available for this option.
logged with access_log_headers = *True.*	No help text available for this option.
reveal_sensitive_prefix = *16*	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.

Configuration option = Default value	Description
	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.
use = *egg:swift#proxy_logging*	Entry point of paste.deploy in the server

Table 10.61. Description of configuration options for [filter-tempauth] in proxy-server.conf

Configuration option = Default value	Description
allow_overrides = *true*	This option allows middleware higher in the WSGI pipeline to override auth processing, useful for middleware such as tempurl and formpost. If you know you are not going to use such middleware and you want a bit of extra security, you can set this to False.
auth_prefix = */auth/*	The HTTP request path prefix for the auth service. Swift itself reserves anything beginning with the letter `v`.
require_group =	No help text available for this option.
reseller_prefix = *AUTH*	The naming scope for the auth service. Swift
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *tempauth*	Label to use when logging
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.
token_life = *86400*	The number of seconds a token is valid.
use = *egg:swift#tempauth*	Entry point of paste.deploy in the server
user_admin_admin = *admin .admin .reseller_admin*	No help text available for this option.
user_test2_tester2 = *testing2 .admin*	No help text available for this option.
user_test5_tester5 = *testing5 service*	No help text available for this option.
user_test_tester = *testing .admin*	No help text available for this option.
user_test_tester3 = *testing3*	No help text available for this option.

Table 10.62. Description of configuration options for [filter-xprofile] in proxy-server.conf

Configuration option = Default value	Description
dump_interval = *5.0*	No help text available for this option.
dump_timestamp = *false*	No help text available for this option.
flush_at_shutdown = *false*	No help text available for this option.
log_filename_prefix = */tmp/log/swift/profile/default.profile*	No help text available for this option.
path = */__profile__*	No help text available for this option.
profile_module = *eventlet.green.profile*	No help text available for this option.
unwind = *false*	No help text available for this option.
use = *egg:swift#xprofile*	Entry point of paste.deploy in the server

Sample proxy server configuration file

```
[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 8080
# 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.  You can withhold subsections by separating the dict level with a
# ".".  The following would cause the sections 'container_quotas' and
 'tempurl'
# to not be listed, and the key max_failed_deletes would be removed from
# bulk_delete.  Default value is 'swift.valid_api_versions' which allows all
# registered features to be listed via HTTP GET /info except
# swift.valid_api_versions information
# disallowed_sections = swift.valid_api_versions, 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
# The following caps the length of log lines to the value given; no limit if
# set to 0, the default.
# log_max_line_length = 0
#
# 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 =
# strict_cors_mode = True
#
# client_timeout = 60
# eventlet_debug = false

[pipeline:main]
pipeline = catch_errors gatekeeper healthcheck proxy-logging cache
 container_sync bulk tempurl ratelimit tempauth container-quotas account-
quotas slo dlo 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 = 65536
# client_chunk_size = 65536
#
# 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 overriden 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-container-meta-temp-url-key, x-container-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
# set log_headers = false
# set log_address = /dev/log
#
# The reseller prefix will verify a token begins with this prefix before even
# attempting to validate it. Also, with authorization, only Swift storage
# accounts with this prefix will be authorized by this middleware. Useful if
# multiple auth systems are in use for one Swift cluster.
# The reseller_prefix may contain a comma separated list of items. The first
# item is used for the token as mentioned above. If second and subsequent
# items exist, the middleware will handle authorization for an account with
# that prefix. For example, for prefixes "AUTH, SERVICE", a path of
# /v1/SERVICE_account is handled the same as /v1/AUTH_account. If an empty
# (blank) reseller prefix is required, it must be first in the list. Two
```

```
# single quote characters indicates an empty (blank) reseller prefix.
# reseller_prefix = AUTH

#
# The require_group parameter names a group that must be presented by
# either X-Auth-Token or X-Service-Token. Usually this parameter is
# used only with multiple reseller prefixes (e.g., SERVICE_require_group=
blah).
# By default, no group is needed. Do not use .admin.
# require_group =

# The auth prefix will cause requests beginning with this prefix to be routed
# to the auth subsystem, for granting tokens, etc.
# auth_prefix = /auth/
# token_life = 86400
#
# This allows middleware higher in the WSGI pipeline to override auth
# processing, useful for middleware such as tempurl and formpost. If you know
# you're not going to use such middleware and you want a bit of extra
 security,
# you can set this to false.
# allow_overrides = true
#
# This specifies what 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.
# storage_url_scheme = default
#
# Lastly, you need to list all the accounts/users you want here. The format
 is:
#   user_<account>_<user> = <key> [group] [group] [...] [storage_url]
# or if you want underscores in <account> or <user>, you can base64 encode
 them
# (with no equal signs) and use this format:
#   user64_<account_b64>_<user_b64> = <key> [group] [group] [...]
 [storage_url]
# There are special groups of:
#   .reseller_admin = can do anything to any account for this auth
#   .admin = can do anything within the account
# If neither of these groups are specified, the user can only access
 containers
# that have been explicitly allowed for them by a .admin or .reseller_admin.
# The trailing optional storage_url allows you to specify an alternate url to
# hand back to the user upon authentication. If not specified, this defaults
 to
# $HOST/v1/<reseller_prefix>_<account> where $HOST will do its best to resolve
# to what the requester would need to use to reach this host.
# Here are example entries, required for running the tests:
user_admin_admin = admin .admin .reseller_admin
user_test_tester = testing .admin
user_test2_tester2 = testing2 .admin
user_test_tester3 = testing3
user_test5_tester5 = testing5 service

# To enable Keystone authentication you need to have the auth token
# middleware first to be configured. Here is an example below, please
# refer to the keystone's documentation for details about the
# different settings.
#
# You'll need to have as well the keystoneauth middleware enabled
```

```
# and have it in your main pipeline so instead of having tempauth in
# there you can change it to: authtoken keystoneauth
#
# [filter:authtoken]
# paste.filter_factory = keystonemiddleware.auth_token:filter_factory
# identity_uri = http://keystonehost:35357/
# auth_uri = http://keystonehost:5000/
# admin_tenant_name = service
# admin_user = swift
# admin_password = password
#
# delay_auth_decision defaults to False, but leaving it as false will
# prevent other auth systems, staticweb, tempurl, formpost, and ACLs from
# working. This value must be explicitly set to True.
# delay_auth_decision = False
#
# cache = swift.cache
# include_service_catalog = False
#
# [filter:keystoneauth]
# use = egg:swift#keystoneauth
# The reseller_prefix option lists account namespaces that this middleware is
# responsible for. The prefix is placed before the Keystone project id.
# For example, for project 12345678, and prefix AUTH, the account is
# named AUTH_12345678 (i.e., path is /v1/AUTH_12345678/...).
# Several prefixes are allowed by specifying a comma-separated list
# as in: "reseller_prefix = AUTH, SERVICE". The empty string indicates a
# single blank/empty prefix. If an empty prefix is required in a list of
# prefixes, a value of '' (two single quote characters) indicates a
# blank/empty prefix. Except for the blank/empty prefix, an underscore ('_')
# character is appended to the value unless already present.
# reseller_prefix = AUTH
#
# The user must have at least one role named by operator_roles on a
# project in order to create, delete and modify containers and objects
# and to set and read privileged headers such as ACLs.
# If there are several reseller prefix items, you can prefix the
# parameter so it applies only to those accounts (for example
# the parameter SERVICE_operator_roles applies to the /v1/SERVICE_<project>
# path). If you omit the prefix, the option applies to all reseller
# prefix items. For the blank/empty prefix, prefix with '' (do not put
# underscore after the two single quote characters).
# operator_roles = admin, swiftoperator
#
# The reseller admin role has the ability to create and delete accounts
# reseller_admin_role = ResellerAdmin
#
# This allows middleware higher in the WSGI pipeline to override auth
# processing, useful for middleware such as tempurl and formpost. If you know
# you're not going to use such middleware and you want a bit of extra
 security,
# you can set this to false.
# allow_overrides = true
#
# If is_admin is true, a user whose username is the same as the project name
# and who has any role on the project will have access rights elevated to be
# the same as if the user had an operator role. Note that the condition
# compares names rather than UUIDs. This option is deprecated.
# is_admin = false
#
```

```
# If the service_roles parameter is present, an X-Service-Token must be
# present in the request that when validated, grants at least one role listed
# in the parameter. The X-Service-Token may be scoped to any project.
# If there are several reseller prefix items, you can prefix the
# parameter so it applies only to those accounts (for example
# the parameter SERVICE_service_roles applies to the /v1/SERVICE_<project>
# path). If you omit the prefix, the option applies to all reseller
# prefix items. For the blank/empty prefix, prefix with '' (do not put
# underscore after the two single quote characters).
# By default, no service_roles are required.
# service_roles =
#
# For backwards compatibility, keystoneauth will match names in cross-tenant
# access control lists (ACLs) when both the requesting user and the tenant
# are in the default domain i.e the domain to which existing tenants are
# migrated. The default_domain_id value configured here should be the same as
# the value used during migration of tenants to keystone domains.
# default_domain_id = default
#
# For a new installation, or an installation in which keystone projects may
# move between domains, you should disable backwards compatible name matching
# in ACLs by setting allow_names_in_acls to false:
# allow_names_in_acls = true

[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".
# This facility may be used to temporarily remove a Swift node from a load
# balancer pool during maintenance or upgrade (remove the file to allow the
# node back into the load balancer pool).
# disable_path =

[filter:cache]
use = egg:swift#memcache
# You can override the default log routing for this filter here:
# set log_name = cache
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log
#
# If not set here, the value for memcache_servers will be read from
# memcache.conf (see memcache.conf-sample) or lacking that file, it will
# default to the value below. You can specify multiple servers separated with
# commas, as in: 10.1.2.3:11211,10.1.2.4:11211
# memcache_servers = 127.0.0.1:11211
#
# Sets how memcache values are serialized and deserialized:
# 0 = older, insecure pickle serialization
# 1 = json serialization but pickles can still be read (still insecure)
# 2 = json serialization only (secure and the default)
# If not set here, the value for memcache_serialization_support will be read
# from /etc/swift/memcache.conf (see memcache.conf-sample).
# To avoid an instant full cache flush, existing installations should
# upgrade with 0, then set to 1 and reload, then after some time (24 hours)
# set to 2 and reload.
# In the future, the ability to use pickle serialization will be removed.
# memcache_serialization_support = 2
#
```

```
# Sets the maximum number of connections to each memcached server per worker
# memcache_max_connections = 2
#
# More options documented in memcache.conf-sample

[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

# DEPRECATED- these will continue to work but will be replaced
# by the X-Account-Sysmeta-Global-Write-Ratelimit flag.
# Please see ratelimiting docs for details.
# 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 POST DELETE
#
# 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: Just needs to be placed before the proxy-server in the pipeline.
[filter:name_check]
use = egg:swift#name_check
# forbidden_chars = '"`<>
# maximum_length = 255
# forbidden_regexp = /\./|/\.\./|/\.$|/\.\.$

[filter:list-endpoints]
use = egg:swift#list_endpoints
# list_endpoints_path = /endpoints/

[filter:proxy-logging]
use = egg:swift#proxy_logging
# If not set, logging directives from [DEFAULT] without "access_" will be used
# access_log_name = swift
# access_log_facility = LOG_LOCAL0
# access_log_level = INFO
# access_log_address = /dev/log
#
# If set, access_log_udp_host will override access_log_address
# access_log_udp_host =
# access_log_udp_port = 514
#
# You can use log_statsd_* from [DEFAULT] or override them here:
# access_log_statsd_host = localhost
# access_log_statsd_port = 8125
# access_log_statsd_default_sample_rate = 1.0
# access_log_statsd_sample_rate_factor = 1.0
# access_log_statsd_metric_prefix =
# access_log_headers = false
#
# If access_log_headers is True and access_log_headers_only is set only
# these headers are logged. Multiple headers can be defined as comma separated
# list like this: access_log_headers_only = Host, X-Object-Meta-Mtime
# access_log_headers_only =
#
# By default, the X-Auth-Token is logged. To obscure the value,
# set reveal_sensitive_prefix to the number of characters to log.
# For example, if set to 12, only the first 12 characters of the
# token appear in the log. An unauthorized access of the log file
# won't allow unauthorized usage of the token. However, the first
# 12 or so characters is unique enough that you can trace/debug
# token usage. Set to 0 to suppress the token completely (replaced
# by '...' in the log).
# Note: reveal_sensitive_prefix will not affect the value
# logged with access_log_headers=True.
# reveal_sensitive_prefix = 16
#
# What HTTP methods are allowed for StatsD logging (comma-sep); request
 methods
```

```
# not in this list will have "BAD_METHOD" for the <verb> portion of the
 metric.
# log_statsd_valid_http_methods = GET,HEAD,POST,PUT,DELETE,COPY,OPTIONS
#
# Note: The double proxy-logging in the pipeline is not a mistake. The
# left-most proxy-logging is there to log requests that were handled in
# middleware and never made it through to the right-most middleware (and
# proxy server). Double logging is prevented for normal requests. See
# proxy-logging docs.

# 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 and staticweb 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 after auth and staticweb in the pipeline.
# 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

# Note: Put after auth in the pipeline.
[filter:container-quotas]
use = egg:swift#container_quotas

# Note: Put after auth in the pipeline.
[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
# Set this to specify this clusters //realm/cluster as "current" in /info
# current = //REALM/CLUSTER

# Note: Put it at the beginning of the pipeline to profile all middleware. But
# it is safer to put this after catch_errors, gatekeeper and healthcheck.
[filter:xprofile]
use = egg:swift#xprofile
# This option enable you to switch profilers which should inherit from python
# standard profiler. Currently the supported value can be 'cProfile',
# 'eventlet.green.profile' etc.
# profile_module = eventlet.green.profile
#
# This prefix will be used to combine process ID and timestamp to name the
# profile data file.  Make sure the executing user has permission to write
# into this path (missing path segments will be created, if necessary).
# If you enable profiling in more than one type of daemon, you must override
# it with an unique value like: /var/log/swift/profile/proxy.profile
# log_filename_prefix = /tmp/log/swift/profile/default.profile
#
# the profile data will be dumped to local disk based on above naming rule
# in this interval.
# dump_interval = 5.0
#
# Be careful, this option will enable profiler to dump data into the file with
# time stamp which means there will be lots of files piled up in the
 directory.
# dump_timestamp = false
#
# This is the path of the URL to access the mini web UI.
# path = /__profile__
#
```

```
# Clear the data when the wsgi server shutdown.
# flush_at_shutdown = false
#
# unwind the iterator of applications
# unwind = false
```

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 10.63. Description of configuration options for [memcache] in memcache.conf

Configuration option = Default value	Description
`connect_timeout` = *0.3*	Timeout in seconds (float) for connection
`io_timeout` = *2.0*	Timeout in seconds (float) for read and write
`memcache_max_connections` = *2*	Max number of connections to each memcached server per worker services
`memcache_serialization_support` = *2*	Sets how memcache values are serialized and deserialized
`memcache_servers` = *127.0.0.1:11211*	Comma-separated list of memcached servers ip:port services
`pool_timeout` = *1.0*	Timeout in seconds (float) for pooled connection
`tries` = *3*	Number of servers to retry on failures getting a pooled connection

Rsyncd configuration

Find an example rsyncd configuration at `etc/rsyncd.conf-sample` in the source code repository.

The available configuration options are:

Table 10.64. Description of configuration options for [account] in rsyncd.conf

Configuration option = Default value	Description
`lock file` = */var/lock/account.lock*	No help text available for this option.
`max connections` = *2*	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.

Table 10.65. Description of configuration options for [container] in rsyncd.conf

Configuration option = Default value	Description
`lock file` = */var/lock/container.lock*	No help text available for this option.
`max connections` = *4*	No help text available for this option.

Configuration option = Default value	Description
path = /srv/node	No help text available for this option.
read only = false	No help text available for this option.

Table 10.66. Description of configuration options for [object] in rsyncd.conf

Configuration option = Default value	Description
lock file = /var/lock/object.lock	No help text available for this option.
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.

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 10.67. Description of configuration options for [filter-ratelimit] in proxy-server.conf

Configuration option = Default value	Description
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,

Configuration option = Default value	Description
	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.
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.
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 rate-limit should be higher than the clock accuracy.
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.
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.
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.
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.
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.
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *ratelimit*	Label to use when logging
use = *egg:swift#ratelimit*	Entry point of paste.deploy in the server
with container_limit_x = *r*	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 10.68. 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 10.69. Description of configuration options for [filter-healthcheck] in account-server.conf

Configuration option = Default value	Description
`disable_path` =	No help text available for this option.
`use` = *egg:swift#healthcheck*	Entry point of paste.deploy in the server

Domain remap

Middleware that translates container and account parts of a domain to path parameters that the proxy server understands.

Table 10.70. Description of configuration options for [filter-domain_remap] in proxy-server.conf

Configuration option = Default value	Description
`path_root` = *v1*	Root path
`reseller_prefixes` = *AUTH*	Reseller prefix
`set log_address` = */dev/log*	Location where syslog sends the logs to
`set log_facility` = *LOG_LOCAL0*	Syslog log facility
`set log_headers` = *false*	If True, log headers in each request
`set log_level` = *INFO*	Log level
`set log_name` = *domain_remap*	Label to use when logging
`storage_domain` = *example.com*	Domain that matches your cloud. Multiple domains can be specified using a comma-separated list.
`use` = *egg:swift#domain_remap*	Entry point of paste.deploy in the server

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 10.71. Description of configuration options for [filter-cname_lookup] in proxy-server.conf

Configuration option = Default value	Description
`lookup_depth` = *1*	Because CNAMES can be recursive, specifies the number of levels through which to search.

Configuration option = Default value	Description
set log_address = */dev/log*	Location where syslog sends the logs to
set log_facility = *LOG_LOCAL0*	Syslog log facility
set log_headers = *false*	If True, log headers in each request
set log_level = *INFO*	Log level
set log_name = *cname_lookup*	Label to use when logging
storage_domain = *example.com*	Domain that matches your cloud. Multiple domains can be specified using a comma-separated list.
use = *egg:swift#cname_lookup*	Entry point of paste.deploy in the server

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
```

If you do not want the object to be downloaded, you can cause `Content-Disposition: inline` to be set on the response by adding the `inline` parameter to the query string, as follows:

```
https://swift-cluster.example.com/v1/AUTH_account/container/object?
temp_url_sig=da39a3ee5e6b4b0d3255bfef95601890afd80709&
temp_url_expires=1323479485&inline
```

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 10.72. Description of configuration options for [filter-tempurl] in proxy-server.conf

Configuration option = Default value	Description
incoming_allow_headers =	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_remove_headers = *x-timestamp*	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.
methods = *GET HEAD PUT POST DELETE*	HTTP methods allowed with Temporary URLs
outgoing_allow_headers = *x-object-meta-public-**	Headers allowed as exceptions to outgoing_allow_headers. Simply a whitespace delimited list of header names and names can optionally end with '*' to indicate a prefix match.
outgoing_remove_headers = *x-object-meta-**	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.
use = *egg:swift#tempurl*	Entry point of paste.deploy in the server

Name check filter

Name Check is a filter that disallows any paths that contain defined forbidden characters or that exceed a defined length.

Table 10.73. Description of configuration options for [filter-name_check] in proxy-server.conf

Configuration option = Default value	Description
forbidden_chars = *'" `<>*	Characters that are not allowed in a name
forbidden_regexp = */\./|/\.\./|/\.$\|/\.\.$*	Substrings to forbid, using regular expression syntax
maximum_length = *255*	Maximum length of a name
use = *egg:swift#name_check*	Entry point of paste.deploy in the server

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 10.74. Description of configuration options for [swift-constraints] in swift.conf

Configuration option = Default value	Description
account_listing_limit = *10000*	The default (and maximum) number of items returned for an account listing request.
container_listing_limit = *10000*	The default (and maximum) number of items returned for a container 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.
max_file_size = *5368709122*	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 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_header_size = *8192*	The max number of bytes in the utf8 encoding of each header. Using 8192 as default because eventlet use 8192 as maximum size of header line. You may need to increase this value when using identity v3 API tokens including more than 7 catalog entries. See also include_service_catalog in proxy-server.conf-sample (documented in overview_auth.rst).
max_meta_count = *90*	The max number of metadata keys that can be stored on a single account, container, or object.
max_meta_name_length = *128*	The max number of bytes in the utf8 encoding of the name portion of a metadata header.
max_meta_overall_size = *4096*	The max number of bytes in the utf8 encoding of the metadata (keys + values).
max_meta_value_length = *256*	The max number of bytes in the utf8 encoding of a metadata value.
max_object_name_length = *1024*	The max number of bytes in the utf8 encoding of an object name.
valid_api_versions = *v0,v1,v2*	No help text available for this option.

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 10.75. Description of configuration options for [dispersion] in dispersion.conf

Configuration option = Default value	Description
auth_key = *testing*	No help text available for this option.
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_version = *1.0*	Indicates which version of auth
concurrency = *25*	Number of replication workers to spawn
container_populate = *yes*	No help text available for this option.
container_report = *yes*	No help text available for this option.
dispersion_coverage = *1.0*	No help text available for this option.
dump_json = *no*	No help text available for this option.
endpoint_type = *publicURL*	Indicates whether endpoint for auth is public or internal
keystone_api_insecure = *no*	Allow accessing insecure keystone server. The keystone's certificate will not be verified.
object_populate = *yes*	No help text available for this option.
object_report = *yes*	No help text available for this option.
retries = *5*	No help text available for this option.
swift_dir = */etc/swift*	Swift configuration directory

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 10.76. Description of configuration options for [filter-slo] in proxy-server.conf

Configuration option = Default value	Description
max_get_time = *86400*	No help text available for this option.
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.

Configuration option = Default value	Description
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
use = *egg:swift#slo*	Entry point of paste.deploy in the server

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 403 response (forbidden).

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 10.77. Description of configuration options for `[filter-container-quotas]` in `proxy-server.conf`

Configuration option = Default value	Description
use = *egg:swift#container_quotas*	Entry point of paste.deploy 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 10.78. Description of configuration options for `[filter-bulk]` in `proxy-server.conf`

Configuration option = Default value	Description
`delete_container_retry_count` = *0*	No help text available for this option.
`max_containers_per_extraction` = *10000*	No help text available for this option.
`max_deletes_per_request` = *10000*	No help text available for this option.
`max_failed_deletes` = *1000*	No help text available for this option.
`max_failed_extractions` = *1000*	No help text available for this option.
`use` = *egg:swift#bulk*	Entry point of paste.deploy in the server
`yield_frequency` = *10*	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
- GET Object
- HEAD Object
- PUT Object
- PUT Object (Copy)

To use this middleware, first download the latest version from its repository to your proxy server(s).

```
$ git clone https://git.openstack.org/cgit/stackforge/swift3/
```

Then, install it using standard python mechanisms, such as:

```
# python setup.py install
```

Alternatively, if you have configured the Ubuntu Cloud Archive, you may use:

```
# apt-get install swift-python-s3
```

To add this middleware to your configuration, add the `swift3` middleware in front of the `swauth` middleware, and before any other middleware that looks at Object Storage requests (like rate limiting).

Ensure that your `proxy-server.conf` file contains `swift3` in the pipeline and the `[filter:swift3]` section, as shown below:

```
[pipeline:main]
pipeline = healthcheck cache swift3 swauth proxy-server

[filter:swift3]
use = egg:swift3#swift3
```

Next, configure the tool that you use to connect to the S3 API. For S3curl, for example, you must add your host IP information by adding your host IP to the @endpoints array (line 33 in s3curl.pl):

```
my @endpoints = ( '1.2.3.4');
```

Now you can send commands to the endpoint, such as:

```
$ ./s3curl.pl - 'a7811544507ebaf6c9a7a8804f47ea1c' -key 'a7d8e981-e296-d2ba-cb3b-db7dd23159bd' -get - -s -v http://1.2.3.4:8080
```

To set up your client, ensure you are using the ec2 credentials, which can be downloaded from the **API Endpoints** tab of the dashboard. The host should also point to the Object Storage node's hostname. It also will have to use the old-style calling format, and not the hostname-based container format. Here is an example client setup using the Python boto library on a locally installed all-in-one Object Storage installation.

```
connection = boto.s3.Connection(
    aws_access_key_id='a7811544507ebaf6c9a7a8804f47ea1c',
    aws_secret_access_key='a7d8e981-e296-d2ba-cb3b-db7dd23159bd',
    port=8080,
    host='127.0.0.1',
    is_secure=False,
    calling_format=boto.s3.connection.OrdinaryCallingFormat())
```

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 10.79. Description of configuration options for `[drive-audit]` in `drive-audit.conf`

Configuration option = Default value	Description
`device_dir` = */srv/node*	Directory devices are mounted under
`error_limit` = *1*	Number of errors to find before a device is unmounted
`log_address` = */dev/log*	Location where syslog sends the logs to
`log_facility` = *LOG_LOCAL0*	Syslog log facility
`log_file_pattern` = */var/log/kern.*[!.][!g][!z]*	Location of the log file with globbing pattern to check against device errors locate device blocks with errors in the log file
`log_level` = *INFO*	Logging level
`log_max_line_length` = *0*	Caps the length of log lines to the value given; no limit if set to 0, the default.
`log_to_console` = *False*	No help text available for this option.
`minutes` = *60*	Number of minutes to look back in `/var/log/kern.log`
`recon_cache_path` = */var/cache/swift*	Directory where stats for a few items will be stored
`regex_pattern_1` = *\berror\b.*\b(dm-[0-9]{1,2}\d?)\b*	No help text available for this option.
`unmount_failed_device` = *True*	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" 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 10.80. Description of configuration options for `[filter-formpost]` in `proxy-server.conf`

Configuration option = Default value	Description
use = *egg:swift#formpost*	Entry point of paste.deploy 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 10.81. Description of configuration options for `[filter-staticweb]` in `proxy-server.conf`

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 that allows code running in a browser (JavaScript for example) to make requests to a domain, other than the one it was originated from. 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
```

New, updated and deprecated options in Kilo for OpenStack Object Storage

Table 10.82. New options

Option = default value	(Type) Help string
container-server.conf: [container-sync] conn_timeout = 5	(StrOpt) Connection timeout to external services
container-server.conf: [container-sync] internal_client_conf_path = /etc/swift/internal-client.conf	(StrOpt) No help text available for this option.
container-server.conf: [container-sync] request_tries = 3	(StrOpt) No help text available for this option.
drive-audit.conf: [drive-audit] log_to_console = False	(StrOpt) No help text available for this option.
drive-audit.conf: [drive-audit] recon_cache_path = /var/cache/swift	(StrOpt) Directory where stats for a few items will be stored
drive-audit.conf: [drive-audit] unmount_failed_device = True	(StrOpt) No help text available for this option.
internal-client.conf: [DEFAULT] log_address = /dev/log	(StrOpt) Location where syslog sends the logs to
internal-client.conf: [DEFAULT] log_custom_handlers =	(StrOpt) Comma-separated list of functions to call to setup custom log handlers.
internal-client.conf: [DEFAULT] log_facility = LOG_LOCAL0	(StrOpt) Syslog log facility
internal-client.conf: [DEFAULT] log_level = INFO	(StrOpt) Logging level
internal-client.conf: [DEFAULT] log_name = swift	(StrOpt) Label used when logging
internal-client.conf: [DEFAULT] log_statsd_default_sample_rate = 1.0	(StrOpt) Defines the probability of sending a sample for any given event or timing measurement.
internal-client.conf: [DEFAULT] log_statsd_host = localhost	(StrOpt) If not set, the StatsD feature is disabled.
internal-client.conf: [DEFAULT] log_statsd_metric_prefix =	(StrOpt) Value will be prepended to every metric sent to the StatsD server.
internal-client.conf: [DEFAULT] log_statsd_port = 8125	(StrOpt) Port value for the StatsD server.
internal-client.conf: [DEFAULT] log_statsd_sample_rate_factor = 1.0	(StrOpt) 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.
internal-client.conf: [DEFAULT] log_udp_host =	(StrOpt) If not set, the UDP receiver for syslog is disabled.
internal-client.conf: [DEFAULT] log_udp_port = 514	(StrOpt) Port value for UDP receiver, if enabled.
internal-client.conf: [DEFAULT] swift_dir = /etc/swift	(StrOpt) Swift configuration directory
internal-client.conf: [DEFAULT] user = swift	(StrOpt) User to run as
internal-client.conf: [app-proxy-server] use = egg:swift#proxy	(StrOpt) Entry point of paste.deploy in the server
internal-client.conf: [filter-cache] use = egg:swift#memcache	(StrOpt) Entry point of paste.deploy in the server
internal-client.conf: [filter-catch_errors] use = egg:swift#catch_errors	(StrOpt) Entry point of paste.deploy in the server
internal-client.conf: [filter-proxy-logging] use = egg:swift#proxy_logging	(StrOpt) Entry point of paste.deploy in the server
internal-client.conf: [pipeline-main] pipeline = catch_errors proxy-logging cache proxy-server	(StrOpt) No help text available for this option.
memcache.conf: [memcache] connect_timeout = 0.3	(StrOpt) Timeout in seconds (float) for connection
memcache.conf: [memcache] io_timeout = 2.0	(StrOpt) Timeout in seconds (float) for read and write
memcache.conf: [memcache] pool_timeout = 1.0	(StrOpt) Timeout in seconds (float) for pooled connection

Option = default value	(Type) Help string
memcache.conf: [memcache] tries = 3	(StrOpt) Number of servers to retry on failures getting a pooled connection
object-server.conf: [object-reconstructor] concurrency = 1	(StrOpt) Number of replication workers to spawn
object-server.conf: [object-reconstructor] daemonize = on	(StrOpt) Whether or not to run replication as a daemon
object-server.conf: [object-reconstructor] handoffs_first = False	(StrOpt) 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.
object-server.conf: [object-reconstructor] http_timeout = 60	(StrOpt) Maximum duration for an HTTP request
object-server.conf: [object-reconstructor] lockup_timeout = 1800	(StrOpt) Attempts to kill all workers if nothing replications for lockup_timeout seconds
object-server.conf: [object-reconstructor] log_address = /dev/log	(StrOpt) Location where syslog sends the logs to
object-server.conf: [object-reconstructor] log_facility = LOG_LOCAL0	(StrOpt) Syslog log facility
object-server.conf: [object-reconstructor] log_level = INFO	(StrOpt) Logging level
object-server.conf: [object-reconstructor] log_name = object-reconstructor	(StrOpt) Label used when logging
object-server.conf: [object-reconstructor] node_timeout = 10	(StrOpt) Request timeout to external services
object-server.conf: [object-reconstructor] reclaim_age = 604800	(StrOpt) Time elapsed in seconds before an object can be reclaimed
object-server.conf: [object-reconstructor] recon_cache_path = /var/cache/swift	(StrOpt) Directory where stats for a few items will be stored
object-server.conf: [object-reconstructor] ring_check_interval = 15	(StrOpt) How often (in seconds) to check the ring
object-server.conf: [object-reconstructor] run_pause = 30	(StrOpt) Time in seconds to wait between replication passes
object-server.conf: [object-reconstructor] stats_interval = 300	(StrOpt) Interval in seconds between logging replication statistics
proxy-server.conf: [filter-authtoken] identity_uri = http://keystonehost:35357/	(StrOpt) No help text available for this option.
proxy-server.conf: [filter-keystoneauth] allow_overrides = true	(StrOpt) This option allows middleware higher in the WSGI pipeline to override auth processing, useful for middleware such as tempurl and formpost. If you know you are not going to use such middleware and you want a bit of extra security, you can set this to False.
proxy-server.conf: [filter-keystoneauth] is_admin = false	(StrOpt) If this option is set to True, it allows to give a user whose username is the same as the project name and who has any role in the project access rights elevated to be the same as if the user had one of the operator_roles. Note that the condition compares names rather than UUIDs. This option is deprecated. It is False by default.
proxy-server.conf: [filter-keystoneauth] reseller_prefix = AUTH	(StrOpt) The naming scope for the auth service. Swift
proxy-server.conf: [filter-keystoneauth] service_roles =	(StrOpt) When present, this option requires that the X-Service-Token header supplies a token from a user who has a role listed in service_roles. This parameter may be prefixed with an appropriate prefix.
proxy-server.conf: [filter-tempauth] require_group =	(StrOpt) No help text available for this option.
proxy-server.conf: [filter-tempauth] user_test5_tester5 = testing5 service	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-0] policy_type = replication	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-1] policy_type = replication	(StrOpt) No help text available for this option.

Option = default value	(Type) Help string
swift.conf: [storage-policy-2] ec_num_data_fragments = 10	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-2] ec_num_parity_fragments = 4	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-2] ec_object_segment_size = 1048576	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-2] ec_type = jerasure_rs_vand	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-2] name = deepfreeze10-4	(StrOpt) No help text available for this option.
swift.conf: [storage-policy-2] policy_type = erasure_coding	(StrOpt) No help text available for this option.
swift.conf: [swift-constraints] valid_api_versions = v0,v1,v2	(StrOpt) No help text available for this option.

Table 10.83. New default values

Option	Previous default value	New default value
object-server.conf: [app-object-server] keep_cache_size	5424880	5242880
proxy-server.conf: [DEFAULT] disallowed_sections	container_quotas, tempurl, bulk_delete.max_failed_deletes	swift.valid_api_versions, container_quotas, tempurl
proxy-server.conf: [app-proxy-server] 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	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-container-meta-temp-url-key, x-container-meta-temp-url-key-2, x-account-access-control
proxy-server.conf: [filter-authtoken] delay_auth_decision	1	False

11. Orchestration

Table of Contents

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 11.1. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
`admin_password` = *None*	(StrOpt) Service user password.
`admin_tenant_name` = *admin*	(StrOpt) Service tenant name.
`admin_token` = *None*	(StrOpt) 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.
`admin_user` = *None*	(StrOpt) Service username.
`auth_admin_prefix` =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
`auth_host` = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_plugin` = *None*	(StrOpt) Name of the plugin to load
`auth_port` = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
`auth_protocol` = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
`auth_section` = *None*	(StrOpt) Config Section from which to load plugin specific options
`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 identity server requires client certificate

Configuration option = Default value	Description
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) 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.
identity_uri = *None*	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = *10*	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = *300*	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = *10*	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = *3*	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = *60*	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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

Configuration option = Default value	Description
	cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = *False*	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = *10*	(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 11.2. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
convergence_engine = *False*	(BoolOpt) Enables engine with convergence architecture. All stacks with this option will be created using convergence engine .
default_deployment_signal_transport = *CFN_SIGNAL*	(StrOpt) Template default for how the server should signal to heat with the deployment output values. CFN_SIGNAL will allow an HTTP POST to a CFN keypair signed URL (requires enabled heat-api-cfn). TEMP_URL_SIGNAL will create a Swift TempURL to be signaled via HTTP PUT (requires object-store endpoint which supports TempURL). HEAT_SIGNAL will allow calls to the Heat API resource-signal using the provided keystone credentials
default_software_config_transport = *POLL_SERVER_CFN*	(StrOpt) Template default for how the server should receive the metadata required for software configuration. POLL_SERVER_CFN will allow calls to the cfn API action DescribeStackResource authenticated with the provided keypair (requires enabled heat-api-cfn). POLL_SERVER_HEAT will allow calls to the Heat API resource-show using the provided keystone credentials (requires keystone v3 API, and configured stack_user_* config options). POLL_TEMP_URL will create and populate a Swift TempURL with metadata for polling (requires object-store endpoint which supports TempURL).
deferred_auth_method = *trusts*	(StrOpt) Select deferred auth method, stored password or trusts.
environment_dir = */etc/heat/environment.d*	(StrOpt) The directory to search for environment files.
error_wait_time = *240*	(IntOpt) Error wait time in seconds for stack action (ie. create or update).
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.
host = *localhost*	(StrOpt) Name of the engine node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address.
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.
memcached_servers = *None*	(ListOpt) Memcached servers or None for in process cache.
periodic_interval = *60*	(IntOpt) Seconds between running periodic tasks.
plugin_dirs = */usr/lib64/heat, /usr/lib/heat, /usr/local/lib/heat, /usr/local/lib64/heat*	(ListOpt) List of directories to search for plug-ins.
[keystone_authtoken]	
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.
[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 11.3. Description of crypt configuration options

Configuration option = Default value	Description
[DEFAULT]	
auth_encryption_key = *notgood but just long enough i t*	(StrOpt) Key used to encrypt authentication info in the database. Length of this key must be 16, 24 or 32 characters.

Table 11.4. Description of database configuration options

Configuration option = Default value	Description
[database]	
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
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 number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Description
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.
slave_connection = None	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = oslo.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 11.5. Description of logging configuration options

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.

Table 11.6. Description of load balancer configuration options

Configuration option = Default value	Description
[DEFAULT]	
loadbalancer_template = None	(StrOpt) Custom template for the built-in loadbalancer nested stack.

Table 11.7. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN	(ListOpt) List of logger=LEVEL pairs.

Configuration option = Default value	Description
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
fatal_exception_format_errors = *False*	(BoolOpt) Make exception message format errors fatal
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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) Enables or disables publication of error events.
syslog_log_facility = *LOG_USER*	(StrOpt) Syslog facility to receive log lines.
use_syslog = *False*	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.
use_stderr = *True*	(BoolOpt) Log output to standard error.
verbose = *False*	(BoolOpt) Print more verbose output (set logging level to INFO instead of default WARNING level).

Table 11.8. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
max_request_body_size = *114688*	(IntOpt) The maximum body size for each request, in bytes.

Table 11.9. Description of quota configuration options

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 = *5*	(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 11.10. Description of Redis configuration options

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 11.11. Description of testing configuration options

Configuration option = Default value	Description
[profiler]	
profiler_enabled = *False*	(BoolOpt) If False fully disable profiling feature.
trace_sqlalchemy = *False*	(BoolOpt) If False do not trace SQL requests.

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 11.12. Description of API configuration options

Configuration option = Default value	Description
[DEFAULT]	
action_retry_limit = *5*	(IntOpt) Number of times to retry to bring a resource to a non-error state. Set to 0 to disable retries.
enable_stack_abandon = *False*	(BoolOpt) Enable the preview Stack Abandon feature.
enable_stack_adopt = *False*	(BoolOpt) Enable the preview Stack Adopt feature.
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.
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.
num_engine_workers = *4*	(IntOpt) Number of heat-engine processes to fork and run.

Configuration option = Default value	Description
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = *['policy.d']*	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched.
policy_file = *policy.json*	(StrOpt) The JSON file that defines policies.
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_scheduler_hints = *False*	(BoolOpt) When this feature is enabled, scheduler hints identifying the heat stack context of a server resource are passed to the configured schedulers in nova, for server creates done using heat resource types OS::Nova::Server and AWS::EC2::Instance. heat_root_stack_id will be set to the id of the root stack of the resource, heat_stack_id will be set to the id of the resource's parent stack, heat_stack_name will be set to the name of the resource's parent stack, heat_path_in_stack will be set to a list of tuples, (stackresourcename, stackname) with list[0] being (None, rootstackname), and heat_resource_name will be set to the resource's name.
stack_user_domain_id = *None*	(StrOpt) Keystone domain ID which contains heat template-defined users. If this option is set, stack_user_domain_name option will be ignored.
stack_user_domain_name = *None*	(StrOpt) Keystone domain name which contains heat template-defined users. If `stack_user_domain_id` option is set, this option is ignored.
trusts_delegated_roles =	(ListOpt) Subset of trustor roles to be delegated to heat. If left unset, all roles of a user will be delegated to heat when creating a stack.
[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.
ca_file = *None*	(StrOpt) Optional CA cert file to use in SSL connections.
cert_file = *None*	(StrOpt) Optional PEM-formatted certificate chain file.
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.
multi_cloud = *False*	(BoolOpt) Allow orchestration of multiple clouds.

Configuration option = Default value	Description
[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.

Table 11.13. Description of Cloudformation-compatible API configuration options

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.

Table 11.14. Description of CloudWatch API configuration options

Configuration option = Default value	Description
[DEFAULT]	
enable_cloud_watch_lite = *True*	(BoolOpt) Enable the legacy OS::Heat::CWLiteAlarm resource.
heat_watch_server_url =	(StrOpt) URL of the Heat CloudWatch server.
[heat_api_cloudwatch]	

Configuration option = Default value	Description
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 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.

Table 11.15. Description of metadata API configuration options

Configuration option = Default value	Description
[DEFAULT]	
heat_metadata_server_url =	(StrOpt) URL of the Heat metadata server.

Table 11.16. Description of waitcondition API configuration options

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 11.17. Description of clients configuration options

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 11.18. Description of client backends configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
`cloud_backend` = *`heat.engine.clients.OpenStackClients`*	(StrOpt) Fully qualified class name to use as a client backend.

Table 11.19. Description of ceilometer clients configuration options

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` = *`None`*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
`insecure` = *`None`*	(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 11.20. Description of cinder clients configuration options

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` = *`None`*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
`http_log_debug` = *`False`*	(BoolOpt) Allow client's debug log output.
`insecure` = *`None`*	(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 11.21. Description of glance clients configuration options

Configuration option = Default value	Description
[clients_glance]	
`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` = *`None`*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
`insecure` = *`None`*	(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 11.22. Description of heat clients configuration options

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` = *`None`*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.

Configuration option = Default value	Description
insecure = *None*	(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 =	(StrOpt) Optional heat url in format like http://0.0.0.0:8004/v1/%(tenant_id)s.

Table 11.23. Description of keystone clients configuration options

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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = *None*	(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 11.24. Description of neutron clients configuration options

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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = *None*	(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 11.25. Description of nova clients configuration options

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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
http_log_debug = *False*	(BoolOpt) Allow client's debug log output.
insecure = *None*	(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 11.26. Description of sahara clients configuration options

Configuration option = Default value	Description
[clients_sahara]	

Configuration option = Default value	Description
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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = *None*	(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 11.27. Description of swift clients configuration options

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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = *None*	(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 11.28. Description of trove clients configuration options

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 = *None*	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = *None*	(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.openstack.common.notifier.rpc_notifier` in the `heat.conf` file:

Table 11.29. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
`amqp_auto_delete` = *False*	(BoolOpt) Auto-delete queues in AMQP.
`amqp_durable_queues` = *False*	(BoolOpt) Use durable queues in AMQP.
`fake_rabbit` = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
`heartbeat_rate` = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
`heartbeat_timeout_threshold` = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
`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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
`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_conn_pool_size` = *30*	(IntOpt) Size of RPC connection pool.

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 11.30. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
`amqp_auto_delete` = *`False`*	(BoolOpt) Auto-delete queues in AMQP.
`amqp_durable_queues` = *`False`*	(BoolOpt) Use durable queues in AMQP.
`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_receiver_capacity` = *`1`*	(IntOpt) The number of prefetched messages held by receiver.
`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 bro-

Configuration option = Default value	Description
	ker 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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

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 11.31. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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 11.32. Description of AMQP configuration options

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.
default_notification_level = *INFO*	(StrOpt) Default notification level for outgoing notifications.
default_publisher_id = *None*	(StrOpt) Default publisher_id for outgoing notifications.
list_notifier_drivers = *None*	(MultiStrOpt) List of drivers to send notifications (DEPRECATED).
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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 11.33. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
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 = 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_response_timeout = 60	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = 64	(IntOpt) Size of RPC thread pool.
[oslo_messaging_amqp]	
allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
container_name = None	(StrOpt) Name for the AMQP container
group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = False	(BoolOpt) Debug: dump AMQP frames to stdout

Table 11.34. Description of notification configuration options

Configuration option = Default value	Description
[DEFAULT]	
onready = None	(StrOpt) Deprecated.

New, updated and deprecated options in Kilo for Orchestration

Table 11.35. New options

Option = default value	(Type) Help string
[DEFAULT] convergence_engine = False	(BoolOpt) Enables engine with convergence architecture. All stacks with this option will be created using convergence engine .
[DEFAULT] default_deployment_signal_transport = CFN_SIGNAL	(StrOpt) Template default for how the server should signal to heat with the deployment output values. CFN_SIGNAL will allow an HTTP POST to a CFN keypair signed URL (requires enabled heat-api-cfn). TEMP_URL_SIGNAL will cre-

Option = default value	(Type) Help string
	ate a Swift TempURL to be signaled via HTTP PUT (requires object-store endpoint which supports TempURL). HEAT_SIGNAL will allow calls to the Heat API resource-signal using the provided keystone credentials
[DEFAULT] default_software_config_transport = POLL_SERVER_CFN	(StrOpt) Template default for how the server should receive the metadata required for software configuration. POLL_SERVER_CFN will allow calls to the cfn API action DescribeStackResource authenticated with the provided keypair (requires enabled heat-api-cfn). POLL_SERVER_HEAT will allow calls to the Heat API resource-show using the provided keystone credentials (requires keystone v3 API, and configured stack_user_* config options). POLL_TEMP_URL will create and populate a Swift TempURL with metadata for polling (requires object-store endpoint which supports TempURL).
[DEFAULT] error_wait_time = 240	(IntOpt) Error wait time in seconds for stack action (ie. create or update).
[DEFAULT] fatal_exception_format_errors = False	(BoolOpt) Make exception message format errors fatal
[DEFAULT] log-config-append = None	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
[DEFAULT] log-date-format = %Y-%m-%d %H:%M:%S	(StrOpt) Format string for %%(asctime)s in log records. Default: %(default)s .
[DEFAULT] log-dir = None	(StrOpt) (Optional) The base directory used for relative –log-file paths.
[DEFAULT] log-file = None	(StrOpt) (Optional) Name of log file to output to. If no default is set, logging will go to stdout.
[DEFAULT] 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.
[DEFAULT] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched.
[DEFAULT] stack_scheduler_hints = False	(BoolOpt) When this feature is enabled, scheduler hints identifying the heat stack context of a server resource are passed to the configured schedulers in nova, for server creates done using heat resource types OS::Nova::Server and AWS::EC2::Instance. heat_root_stack_id will be set to the id of the root stack of the resource, heat_stack_id will be set to the id of the resource's parent stack, heat_stack_name will be set to the name of the resource's parent stack, heat_path_in_stack will be set to a list of tuples, (stackresourcename, stackname) with list[0] being (None, rootstackname), and heat_resource_name will be set to the resource's name.
[DEFAULT] syslog-log-facility = LOG_USER	(StrOpt) Syslog facility to receive log lines.
[DEFAULT] use-syslog = False	(BoolOpt) Use syslog for logging. Existing syslog format is DEPRECATED during I, and will change in J to honor RFC5424.
[DEFAULT] use-syslog-rfc-format = False	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the APP-NAME is deprecated in I, and will be removed in J.

Option = default value	(Type) Help string
[clients_sahara] ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
[clients_sahara] cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
[clients_sahara] endpoint_type = None	(StrOpt) Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
[clients_sahara] insecure = None	(BoolOpt) If set, then the server's certificate will not be verified.
[clients_sahara] key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
[ec2authtoken] ca_file = None	(StrOpt) Optional CA cert file to use in SSL connections.
[ec2authtoken] cert_file = None	(StrOpt) Optional PEM-formatted certificate chain file.
[ec2authtoken] insecure = False	(BoolOpt) If set, then the server's certificate will not be verified.
[ec2authtoken] key_file = None	(StrOpt) Optional PEM-formatted file that contains the private key.
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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

Option = default value	(Type) Help string
	when they are able to take everything down, as it requires a clean break.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.
[profiler] profiler_enabled = False	(BoolOpt) If False fully disable profiling feature.
[profiler] trace_sqlalchemy = False	(BoolOpt) If False do not trace SQL requests.

Table 11.36. New default values

Option	Previous default value	New default value
[DEFAULT] auth_encryption_key	notgood but just long enough i think	notgood but just long enough i t
[DEFAULT] default_log_levels	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN	amqp=WARN, amqplib=WARN, boto=WARN, qpid=WARN, sqlalchemy=WARN, suds=INFO, oslo.messaging=INFO, iso8601=WARN, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, requests.packages.urllib3.util.retry=WARN, urllib3.util.retry=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN
[DEFAULT] deferred_auth_method	password	trusts
[DEFAULT] max_nested_stack_depth	3	5
[DEFAULT] num_engine_workers	1	4
[DEFAULT] plugin_dirs	/usr/lib64/heat, /usr/lib/heat	/usr/lib64/heat, /usr/lib/heat, /usr/local/lib/heat, /usr/local/lib64/heat
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost	local
[DEFAULT] trusts_delegated_roles	heat_stack_owner	

Table 11.37. Deprecated options

Deprecated option	New Option
[DEFAULT] log-format	None
[DEFAULT] use-syslog	None
[DEFAULT] list_notifier_drivers	None

12. Telemetry

Table of Contents

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 12.1. Description of alarm configuration options

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` = *default*	(StrOpt) Driver to use for alarm evaluation service. DEPRECATED: "singleton" and "partitioned" alarm evaluator services will be removed in Kilo in favour of the default alarm evaluation service using tooz for partitioning.
`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. DEPRECATED: RPC-based partitionedalarm evaluation service will be removed in Kilo in favour of the default alarm evaluation service using tooz for partitioning.
`project_alarm_quota` = *None*	(IntOpt) Maximum number of alarms defined for a project.
`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_max_retries` = *0*	(IntOpt) Number of retries for REST notifier
`rest_notifier_ssl_verify` = *True*	(BoolOpt) Whether to verify the SSL Server certificate when calling alarm action.
`user_alarm_quota` = *None*	(IntOpt) Maximum number of alarms defined for a user.

Table 12.2. Description of alarms configuration options

Configuration option = Default value	Description
[alarms]	
`gnocchi_url` = *http://localhost:8041*	(StrOpt) URL to Gnocchi.

Table 12.3. Description of AMQP configuration options

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.
notification_driver = *[]*	(MultiStrOpt) Driver or drivers to handle sending notifications.
notification_topics = *notifications*	(ListOpt) AMQP topic used for OpenStack notifications.
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 12.4. Description of API configuration options

Configuration option = Default value	Description
[DEFAULT]	
api_paste_config = *api_paste.ini*	(StrOpt) Configuration file for WSGI definition of API.
api_workers = *1*	(IntOpt) Number of workers for Ceilometer API server.
event_pipeline_cfg_file = *event_pipeline.yaml*	(StrOpt) Configuration file for event pipeline definition.
pipeline_cfg_file = *pipeline.yaml*	(StrOpt) Configuration file for pipeline definition.
reserved_metadata_keys =	(ListOpt) List of metadata keys reserved for metering use. And these keys are additional to the ones included in the namespace.
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.
pecan_debug = *False*	(BoolOpt) Toggle Pecan Debug Middleware.
port = *8777*	(IntOpt) The port for the ceilometer API server.

Table 12.5. Description of authorization configuration options

Configuration option = Default value	Description
[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 12.6. Description of authorization token configuration options

Configuration option = Default value	Description
[keystone_authtoken]	
admin_password = *None*	(StrOpt) Service user password.
admin_tenant_name = *admin*	(StrOpt) Service tenant name.
admin_token = *None*	(StrOpt) 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.
admin_user = *None*	(StrOpt) Service username.
auth_admin_prefix =	(StrOpt) Prefix to prepend at the beginning of the path. Deprecated, use identity_uri.
auth_host = *127.0.0.1*	(StrOpt) Host providing the admin Identity API endpoint. Deprecated, use identity_uri.
auth_plugin = *None*	(StrOpt) Name of the plugin to load
auth_port = *35357*	(IntOpt) Port of the admin Identity API endpoint. Deprecated, use identity_uri.
auth_protocol = *https*	(StrOpt) Protocol of the admin Identity API endpoint (http or https). Deprecated, use identity_uri.
auth_section = *None*	(StrOpt) Config Section from which to load plugin specific options
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 identity server requires client certificate
check_revocations_for_cached = *False*	(BoolOpt) If true, the revocation list will be checked for cached tokens. This requires that PKI tokens are configured on the identity server.
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.
hash_algorithms = *md5*	(ListOpt) 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.
http_connect_timeout = *None*	(IntOpt) Request timeout value for communicating with Identity API server.

Configuration option = Default value	Description
http_request_max_retries = 3	(IntOpt) How many times are we trying to reconnect when communicating with Identity API Server.
identity_uri = None	(StrOpt) Complete admin Identity API endpoint. This should specify the unversioned root endpoint e.g. https://localhost:35357/
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 identity server requires client certificate
memcache_pool_conn_get_timeout = 10	(IntOpt) (Optional) Number of seconds that an operation will wait to get a memcache client connection from the pool.
memcache_pool_dead_retry = 300	(IntOpt) (Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	(IntOpt) (Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	(IntOpt) (Optional) Socket timeout in seconds for communicating with a memcache server.
memcache_pool_unused_timeout = 60	(IntOpt) (Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
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.
memcache_use_advanced_pool = False	(BoolOpt) (Optional) Use the advanced (eventlet safe) memcache client pool. The advanced pool will only work under python 2.x.
revocation_cache_time = 10	(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 12.7. Description of collector configuration options

Configuration option = Default value	Description
[DEFAULT]	
collector_workers = 1	(IntOpt) Number of workers for collector service. A single collector is enabled by default.
[collector]	

Configuration option = Default value	Description
requeue_event_on_dispatcher_error = False	(BoolOpt) Requeue the event on the collector event queue when the collector fails to dispatch it.
requeue_sample_on_dispatcher_error = False	(BoolOpt) Requeue the sample on the collector sample queue when the collector fails to dispatch it. This is only valid if the sample come from the notifier publisher.
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 12.8. Description of common configuration options

Configuration option = Default value	Description
[DEFAULT]	
host = localhost	(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.
http_timeout = 600	(IntOpt) Timeout seconds for HTTP requests. Set it to None to disable timeout.
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.
polling_namespaces = ['compute', 'central']	(MultiChoicesOpt) Polling namespace(s) to be used while resource polling
pollster_list = []	(MultiChoicesOpt) List of pollsters (or wildcard templates) to be used while polling
rootwrap_config = /etc/ceilometer/rootwrap.conf	(StrOpt) Path to the rootwrap configuration file touse for running commands as root
shuffle_time_before_polling_task = 0	(IntOpt) To reduce large requests at same time to Nova or other components from different compute agents, shuffle start time of polling task.
sql_expire_samples_only = False	(BoolOpt) Indicates if expirer expires only samples. If set true, expired samples will be deleted, but residual resource and meter definition data will remain.
[compute]	
workload_partitioning = False	(BoolOpt) Enable work-load partitioning, allowing multiple compute agents to be run simultaneously.
[coordination]	
backend_url = None	(StrOpt) The backend URL to use for distributed coordination. If left empty, per-deployment central agent and per-host compute agent won't do workload partitioning and will only function correctly if a single instance of that service is running.
check_watchers = 10.0	(FloatOpt) Number of seconds between checks to see if group membership has changed
heartbeat = 1.0	(FloatOpt) Number of seconds between heartbeats for distributed coordination.
[keystone_authtoken]	

Configuration option = Default value	Description
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.
[polling]	
partitioning_group_prefix = *None*	(StrOpt) Work-load partitioning group prefix. Use only if you want to run multiple polling agents with different config files. For each sub-group of the agent pool with the same partitioning_group_prefix a disjoint subset of pollsters should be loaded.

Table 12.9. Description of concurrency configuration options

Configuration option = Default value	Description
[oslo_concurrency]	
disable_process_locking = *False*	(BoolOpt) Enables or disables inter-process locks.
lock_path = *None*	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.

Table 12.10. Description of database configuration options

Configuration option = Default value	Description
[DEFAULT]	
database_connection = *None*	(StrOpt) DEPRECATED - Database connection string.
[database]	
alarm_connection = *None*	(StrOpt) The connection string used to connect to the alarm database. (if unset, connection is used)
backend = *sqlalchemy*	(StrOpt) The back end to use for the database.
connection = *None*	(StrOpt) The SQLAlchemy connection string to use 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.
db2nosql_resource_id_maxlen = *512*	(IntOpt) The max length of resources id in DB2 nosql, the value should be larger than len(hostname) * 2 as compute node's resource id is <hostname>_<nodename>.
db_inc_retry_interval = *True*	(BoolOpt) If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = *20*	(IntOpt) Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.
db_max_retry_interval = *10*	(IntOpt) If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = *1*	(IntOpt) Seconds between retries of a database transaction.
event_connection = *None*	(StrOpt) The connection string used to connect to the event database. (if unset, connection is used)
event_time_to_live = *-1*	(IntOpt) Number of seconds that events are kept in the database for (<= 0 means forever).
idle_timeout = *3600*	(IntOpt) Timeout before idle SQL connections are reaped.

Configuration option = Default value	Description
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 number of database connection retries during startup. Set to -1 to specify an infinite retry count.
metering_connection = *None*	(StrOpt) The connection string used to connect to the metering database. (if unset, connection is used)
metering_time_to_live = *-1*	(IntOpt) Number of seconds that samples are kept in the database for (<= 0 means forever).
min_pool_size = *1*	(IntOpt) Minimum number of SQL connections to keep open in a pool.
mongodb_replica_set =	(StrOpt) The name of the replica set which is used to connect to MongoDB database. If it is set, MongoReplicaSetClient will be used instead of MongoClient.
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.
slave_connection = *None*	(StrOpt) The SQLAlchemy connection string to use to connect to the slave database.
sqlite_db = *oslo.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 12.11. Description of logging configuration options

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.
nova_http_log_debug = *False*	(BoolOpt) Allow novaclient's debug log output.

Table 12.12. Description of HTTP dispatcher configuration options

Configuration option = Default value	Description
[dispatcher_http]	
cadf_only = *False*	(BoolOpt) The flag that indicates if only cadf message should be posted. If false, all meters will be posted.
event_target = *None*	(StrOpt) The target for event data where the http request will be sent to. If this is not set, it will default to same as Sample target.

Configuration option = Default value	Description
target =	(StrOpt) The target where the http request will be sent. If this is not set, no data will be posted. For example: target = http://hostname:1234/path
timeout = 5	(IntOpt) The max time in seconds to wait for a request to timeout.

Table 12.13. Description of events configuration options

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)
store_raw = []	(MultiStrOpt) Store the raw notification for select priority levels (info and/or error). By default, raw details are not captured.
[notification]	
ack_on_event_error = True	(BoolOpt) Acknowledge message when event persistence fails.
store_events = False	(BoolOpt) Save event details.
workload_partitioning = False	(BoolOpt) Enable workload partitioning, allowing multiple notification agents to be run simultaneously.

Table 12.14. Description of exchange configuration options

Configuration option = Default value	Description
[DEFAULT]	
cinder_control_exchange = cinder	(StrOpt) Exchange name for Cinder notifications.
glance_control_exchange = glance	(StrOpt) Exchange name for Glance notifications.
heat_control_exchange = heat	(StrOpt) Exchange name for Heat notifications
http_control_exchanges = ['nova', 'glance', 'neutron', 'cinder']	(MultiStrOpt) Exchanges name to listen for notifications.
ironic_exchange = ironic	(StrOpt) Exchange name for Ironic notifications.
keystone_control_exchange = keystone	(StrOpt) Exchange name for Keystone notifications.
neutron_control_exchange = neutron	(StrOpt) Exchange name for Neutron notifications.
nova_control_exchange = nova	(StrOpt) Exchange name for Nova notifications.
sahara_control_exchange = sahara	(StrOpt) Exchange name for Data Processing notifications.
sample_source = openstack	(StrOpt) Source for samples emitted on this instance.
swift_control_exchange = swift	(StrOpt) Exchange name for Swift notifications.
trove_control_exchange = trove	(StrOpt) Exchange name for DBaaS notifications.

Table 12.15. Description of glance configuration options

Configuration option = Default value	Description
[DEFAULT]	
glance_page_size = 0	(IntOpt) Number of items to request in each paginated Glance API request (parameter used by glancecelient). If this is less than or equal to 0, page size is not specified (default value in glanceclient is used).

Table 12.16. Description of inspector configuration options

Configuration option = Default value	Description
[DEFAULT]	
hypervisor_inspector = *libvirt*	(StrOpt) Inspector to use for inspecting the hypervisor layer.
libvirt_type = *kvm*	(StrOpt) Libvirt domain type.
libvirt_uri =	(StrOpt) Override the default libvirt URI (which is dependent on libvirt_type).

Table 12.17. Description of IPMI configuration options

Configuration option = Default value	Description
[ipmi]	
node_manager_init_retry = *3*	(IntOpt) Number of retries upon Intel Node Manager initialization failure
polling_retry = *3*	(IntOpt) Tolerance of IPMI/NM polling failures before disable this pollster. Negative indicates retrying forever.

Table 12.18. Description of oslo_middleware configuration options

Configuration option = Default value	Description
[oslo_middleware]	
max_request_body_size = *114688*	(IntOpt) The maximum body size for each request, in bytes.

Table 12.19. Description of logging configuration options

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, requests.packages.urllib3.connectionpool=WARN, urllib3.connectionpool=WARN, websocket=WARN, keystonemiddleware=WARN, routes.middleware=WARN, stevedore=WARN*	(ListOpt) List of logger=LEVEL pairs.
fatal_deprecations = *False*	(BoolOpt) Enables or disables fatal status of deprecations.
instance_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance that is passed with the log message.
instance_uuid_format = *"[instance: %(uuid)s] "*	(StrOpt) The format for an instance UUID that is passed with the log message.
log_config_append = *None*	(StrOpt) The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation.
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.

Configuration option = Default value	Description
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) Enables or disables publication of 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 will change in J to honor RFC5424.
use_syslog_rfc_format = *False*	(BoolOpt) (Optional) Enables or disables syslog rfc5424 format for logging. If enabled, prefixes the MSG part of the syslog message with APP-NAME (RFC5424). The format without the 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 12.20. Description of MagnetoDB configuration options

Configuration option = Default value	Description
[DEFAULT]	
magnetodb_control_exchange = *magnetodb*	(StrOpt) Exchange name for Magnetodb notifications.

Table 12.21. Description of notification configuration options

Configuration option = Default value	Description
[notification]	
disable_non_metric_meters = *False*	(BoolOpt) WARNING: Ceilometer historically offered the ability to store events as meters. This usage is NOT advised as it can flood the metering database and cause performance degradation. This option disables the collection of non-metric meters and will be the default behavior in Liberty.

Table 12.22. Description of policy configuration options

Configuration option = Default value	Description
[oslo_policy]	
policy_default_rule = *default*	(StrOpt) Default rule. Enforced when a requested rule is not found.
policy_dirs = *['policy.d']*	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the

Configuration option = Default value	Description
	search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = *policy.json*	(StrOpt) The JSON file that defines policies.

Table 12.23. Description of Qpid configuration options

Configuration option = Default value	Description
[oslo_messaging_qpid]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
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_receiver_capacity = *1*	(IntOpt) The number of prefetched messages held by receiver.
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.
rpc_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 12.24. Description of RabbitMQ configuration options

Configuration option = Default value	Description
[oslo_messaging_rabbit]	
amqp_auto_delete = *False*	(BoolOpt) Auto-delete queues in AMQP.
amqp_durable_queues = *False*	(BoolOpt) Use durable queues in AMQP.
fake_rabbit = *False*	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
heartbeat_rate = *2*	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = *0*	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL
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).

Configuration option = Default value	Description
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 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 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.
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_conn_pool_size = *30*	(IntOpt) Size of RPC connection pool.

Table 12.25. Description of Redis configuration options

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 12.26. Description of Rados gateway configuration options

Configuration option = Default value	Description
[rgw_admin_credentials]	
access_key = *None*	(StrOpt) Access key for Radosgw Admin.
secret_key = *None*	(StrOpt) Secret key for Radosgw Admin.

Table 12.27. Description of RPC configuration options

Configuration option = Default value	Description
[DEFAULT]	
dispatcher = *['database']*	(MultiStrOpt) Dispatcher to process data.
matchmaker_heartbeat_freq = *300*	(IntOpt) Heartbeat frequency.
matchmaker_heartbeat_ttl = *600*	(IntOpt) Heartbeat time-to-live.

Configuration option = Default value	Description
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_response_timeout = *60*	(IntOpt) Seconds to wait for a response from a call.
rpc_thread_pool_size = *64*	(IntOpt) Size of RPC thread pool.
[notification]	
messaging_urls = *[]*	(MultiStrOpt) Messaging URLs to listen for notifications. Example: transport://user:pass@host1:port[,hostN:portN]/virtual_host (DEFAULT/transport_url is used if empty)
[oslo_messaging_amqp]	
allow_insecure_clients = *False*	(BoolOpt) Accept clients using either SSL or plain TCP
broadcast_prefix = *broadcast*	(StrOpt) address prefix used when broadcasting to all servers
container_name = *None*	(StrOpt) Name for the AMQP container
group_request_prefix = *unicast*	(StrOpt) address prefix when sending to any server in group
idle_timeout = *0*	(IntOpt) Timeout for inactive connections (in seconds)
server_request_prefix = *exclusive*	(StrOpt) address prefix used when sending to a specific server
ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
ssl_key_password = *None*	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
trace = *False*	(BoolOpt) Debug: dump AMQP frames to stdout
[publisher]	
telemetry_secret = *change this for valid signing*	(StrOpt) Secret value for signing messages. Set value empty if signing is not required to avoid computational overhead.
[publisher_notifier]	
event_topic = *event*	(StrOpt) The topic that ceilometer uses for event notifications.
metering_topic = *metering*	(StrOpt) The topic that ceilometer uses for metering notifications.
telemetry_driver = *messagingv2*	(StrOpt) The driver that ceilometer uses for metering notifications.
[publisher_rpc]	
metering_topic = *metering*	(StrOpt) The topic that ceilometer uses for metering messages.

Table 12.28. Description of service types configuration options

Configuration option = Default value	Description
[service_types]	
glance = *image*	(StrOpt) Glance service type.
kwapi = *energy*	(StrOpt) Kwapi service type.
neutron = *network*	(StrOpt) Neutron service type.
nova = *compute*	(StrOpt) Nova service type.

Configuration option = Default value	Description
radosgw = *object-store*	(StrOpt) Radosgw service type.
swift = *object-store*	(StrOpt) Swift service type.

Table 12.29. Description of swift configuration options

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 12.30. Description of TripleO configuration options

Configuration option = Default value	Description
[hardware]	
readonly_user_name = *ro_snmp_user*	(StrOpt) SNMPd user name of all nodes running in the cloud.
readonly_user_password = *password*	(StrOpt) SNMPd password of all the nodes running in the cloud.
url_scheme = *snmp://*	(StrOpt) URL scheme to use for hardware nodes.

Table 12.31. Description of VMware configuration options

Configuration option = Default value	Description
[vmware]	
api_retry_count = *10*	(IntOpt) Number of times a VMware Vsphere API may be retried.
host_ip =	(StrOpt) IP address of the VMware Vsphere host.
host_password =	(StrOpt) Password of VMware Vsphere.
host_port = *443*	(IntOpt) Port of the VMware Vsphere host.
host_username =	(StrOpt) Username of VMware Vsphere.
task_poll_interval = *0.5*	(FloatOpt) Sleep time in seconds for polling an ongoing async task.
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 12.32. Description of XenAPI configuration options

Configuration option = Default value	Description
[xenapi]	
connection_password = *None*	(StrOpt) Password for connection to XenServer/Xen Cloud Platform.
connection_url = *None*	(StrOpt) URL for connection to XenServer/Xen Cloud Platform.
connection_username = *root*	(StrOpt) Username for connection to XenServer/Xen Cloud Platform.
login_timeout = *10*	(IntOpt) Timeout in seconds for XenAPI login.

Table 12.33. Description of Zaqar configuration options

Configuration option = Default value	Description
[DEFAULT]	

Configuration option = Default value	Description
zaqar_control_exchange = *zaqar*	(StrOpt) Exchange name for Messaging service notifications.

Table 12.34. Description of ZeroMQ configuration options

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 = *localhost*	(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 = *local*	(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.

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]

#
# From ceilometer
#

# To reduce large requests at same time to Nova or other components
# from different compute agents, shuffle start time of polling task.
# (integer value)
#shuffle_time_before_polling_task = 0

# Configuration file for WSGI definition of API. (string value)
#api_paste_config = api_paste.ini

# Number of workers for Ceilometer API server. (integer value)
#api_workers = 1

# Polling namespace(s) to be used while resource polling (unknown
# type)
#polling_namespaces = ['compute', 'central']

# List of pollsters (or wildcard templates) to be used while polling
```

```
# (unknown type)
#pollster_list = []

# Exchange name for Nova notifications. (string value)
#nova_control_exchange = nova

# 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

# List of metadata keys reserved for metering use. And these keys are
# additional to the ones included in the namespace. (list value)
#reserved_metadata_keys =

# Inspector to use for inspecting the hypervisor layer. (string value)
#hypervisor_inspector = libvirt

# Libvirt domain type. (string value)
# Allowed values: kvm, lxc, qemu, uml, xen
#libvirt_type = kvm

# Override the default libvirt URI (which is dependent on
# libvirt_type). (string value)
#libvirt_uri =

# Exchange name for Data Processing notifications. (string value)
#sahara_control_exchange = sahara

# Dispatcher to process data. (multi valued)
# Deprecated group/name - [collector]/dispatcher
#dispatcher = database

# Exchange name for Keystone notifications. (string value)
#keystone_control_exchange = keystone

# Number of items to request in each paginated Glance API request
# (parameter used by glancecelient). If this is less than or equal to
# 0, page size is not specified (default value in glanceclient is
# used). (integer value)
#glance_page_size = 0

# Exchange name for Glance notifications. (string value)
#glance_control_exchange = glance

# Exchange name for Ironic notifications. (string value)
#ironic_exchange = ironic

# Exchanges name to listen for notifications. (multi valued)
#http_control_exchanges = nova
#http_control_exchanges = glance
#http_control_exchanges = neutron
#http_control_exchanges = cinder

# Exchange name for Neutron notifications. (string value)
# Deprecated group/name - [DEFAULT]/quantum_control_exchange
#neutron_control_exchange = neutron

# Allow novaclient's debug log output. (boolean value)
```

```
#nova_http_log_debug = false

# Swift reseller prefix. Must be on par with reseller_prefix in proxy-
# server.conf. (string value)
#reseller_prefix = AUTH_

# 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>

# 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

# The name of a logging configuration file. This file is appended to
# any existing logging configuration files. For details about logging
# configuration files, see the Python logging module documentation.
# (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 will change in J to honor RFC5424. (boolean value)
#use_syslog = false

# (Optional) Enables or disables syslog rfc5424 format for logging. If
# enabled, prefixes the MSG part of the syslog message with APP-NAME
# (RFC5424). The format without the 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

# 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,amqplib=WARN,boto=WARN,qpid=WARN,sqlalchemy=
WARN,suds=INFO,oslo.messaging=INFO,iso8601=WARN,requests.packages.
urllib3.connectionpool=WARN,urllib3.connectionpool=WARN,websocket=WARN,
keystonemiddleware=WARN,routes.middleware=WARN,stevedore=WARN

# Enables or disables publication of error events. (boolean value)
#publish_errors = false

# Enables or disables fatal status of deprecations. (boolean value)
#fatal_deprecations = false

# The format for an instance that is passed with the log message.
# (string value)
#instance_format = "[instance: %(uuid)s] "

# The format for an instance UUID that is passed with the log message.
# (string value)
#instance_uuid_format = "[instance: %(uuid)s] "

# Exchange name for Heat notifications (string value)
#heat_control_exchange = heat

# Configuration file for pipeline definition. (string value)
#pipeline_cfg_file = pipeline.yaml

# Configuration file for event pipeline definition. (string value)
#event_pipeline_cfg_file = event_pipeline.yaml

# Exchange name for DBaaS notifications. (string value)
#trove_control_exchange = trove

# Exchange name for Messaging service notifications. (string value)
#zaqar_control_exchange = zaqar

# Source for samples emitted on this instance. (string value)
```

```
# Deprecated group/name - [DEFAULT]/counter_source
#sample_source = openstack

# 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 = shock

# 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

# Timeout seconds for HTTP requests. Set it to None to disable
# timeout. (integer value)
#http_timeout = 600

# DEPRECATED - Database connection string. (string value)
#database_connection = <None>

# Path to the rootwrap configuration file touse for running commands
# as root (string value)
#rootwrap_config = /etc/ceilometer/rootwrap.conf

# Exchange name for Cinder notifications. (string value)
#cinder_control_exchange = cinder

#
# From oslo.messaging
#

# 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 = local

# 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 = localhost

# 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 thread 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


[alarm]

#
# From ceilometer
#

# 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

# Number of retries for REST notifier (integer value)
#rest_notifier_max_retries = 0

# Period of evaluation cycle, should be >= than configured pipeline
# interval for collection of underlying meters. (integer value)
# Deprecated group/name - [alarm]/threshold_evaluation_interval
#evaluation_interval = 60
```

```
# 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. DEPRECATED: RPC-based partitionedalarm evaluation service
# will be removed in Kilo in favour of the default alarm evaluation
# service using tooz for partitioning. (string value)
#partition_rpc_topic = alarm_partition_coordination

# URL to Gnocchi. (string value)
#gnocchi_url = http://localhost:8041

# Record alarm change events. (boolean value)
#record_history = true

# Maximum number of alarms defined for a user. (integer value)
#user_alarm_quota = <None>

# Maximum number of alarms defined for a project. (integer value)
#project_alarm_quota = <None>

# Driver to use for alarm evaluation service. DEPRECATED: "singleton"
# and "partitioned" alarm evaluator services will be removed in Kilo
# in favour of the default alarm evaluation service using tooz for
# partitioning. (string value)
#evaluation_service = default


[api]

#
# From ceilometer
#

# 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

# Toggle Pecan Debug Middleware. (boolean value)
#pecan_debug = false


[central]

#
# From ceilometer
#

# Work-load partitioning group prefix. Use only if you want to run
# multiple polling agents with different config files. For each sub-
# group of the agent pool with the same partitioning_group_prefix a
# disjoint subset of pollsters should be loaded. (string value)
# Deprecated group/name - [central]/partitioning_group_prefix
#partitioning_group_prefix = <None>
```

```
[collector]

#
# From ceilometer
#

# 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

# Requeue the sample on the collector sample queue when the collector
# fails to dispatch it. This is only valid if the sample come from the
# notifier publisher. (boolean value)
#requeue_sample_on_dispatcher_error = false

# Requeue the event on the collector event queue when the collector
# fails to dispatch it. (boolean value)
#requeue_event_on_dispatcher_error = false


[compute]

#
# From ceilometer
#

# Enable work-load partitioning, allowing multiple compute agents to
# be run simultaneously. (boolean value)
#workload_partitioning = false


[coordination]

#
# From ceilometer
#

# The backend URL to use for distributed coordination. If left empty,
# per-deployment central agent and per-host compute agent won't do
# workload partitioning and will only function correctly if a single
# instance of that service is running. (string value)
#backend_url = <None>

# Number of seconds between heartbeats for distributed coordination.
# (floating point value)
#heartbeat = 1.0

# Number of seconds between checks to see if group membership has
# changed (floating point value)
#check_watchers = 10.0


[database]

#
# From ceilometer
```

```
#

# Number of seconds that samples are kept in the database for (<= 0
# means forever). (integer value)
# Deprecated group/name - [database]/time_to_live
#metering_time_to_live = -1

# Number of seconds that events are kept in the database for (<= 0
# means forever). (integer value)
#event_time_to_live = -1

# The connection string used to connect to the metering database. (if
# unset, connection is used) (string value)
#metering_connection = <None>

# The connection string used to connect to the alarm database. (if
# unset, connection is used) (string value)
#alarm_connection = <None>

# The connection string used to connect to the event database. (if
# unset, connection is used) (string value)
#event_connection = <None>

# The name of the replica set which is used to connect to MongoDB
# database. If it is set, MongoReplicaSetClient will be used instead
# of MongoClient. (string value)
#mongodb_replica_set =

# The max length of resources id in DB2 nosql, the value should be
# larger than len(hostname) * 2 as compute node's resource id is
# <hostname>_<nodename>. (integer value)
#db2nosql_resource_id_maxlen = 512

#
# From oslo.db
#

# The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
#sqlite_db = oslo.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use 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 SQLAlchemy connection string to use to connect to the slave
# database. (string value)
#slave_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 number of database connection retries during startup. Set to
# -1 to specify 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 retries of a database transaction. (integer value)
#db_retry_interval = 1

# If True, increases the interval between retries of a database
# operation up to db_max_retry_interval. (boolean value)
#db_inc_retry_interval = true

# If db_inc_retry_interval is set, the maximum seconds between retries
# of a database operation. (integer value)
#db_max_retry_interval = 10

# Maximum retries in case of connection error or deadlock error before
# error is raised. Set to -1 to specify an infinite retry count.
# (integer value)
#db_max_retries = 20


[dispatcher_file]

#
# From ceilometer
#

# 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]

#
# From ceilometer
#

# 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

# Store the raw notification for select priority levels (info and/or
# error). By default, raw details are not captured. (multi valued)
#store_raw =


[hardware]

#
# From ceilometer
#

# URL scheme to use for hardware nodes. (string value)
#url_scheme = snmp://
```

```
# SNMPd user name of all nodes running in the cloud. (string value)
#readonly_user_name = ro_snmp_user

# SNMPd password of all the nodes running in the cloud. (string value)
#readonly_user_password = password


[ipmi]

#
# From ceilometer
#

# Number of retries upon Intel Node Manager initialization failure
# (integer value)
#node_manager_init_retry = 3

# Tolerance of IPMI/NM polling failures before disable this pollster.
# Negative indicates retrying forever. (integer value)
#polling_retry = 3


[keystone_authtoken]

#
# From keystonemiddleware.auth_token
#

# 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.
# (integer 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

# Env key for the swift cache. (string value)
#cache = <None>

# Required if identity server requires client certificate (string
# value)
#certfile = <None>

# Required if identity 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) Number of seconds memcached server is considered dead
# before it is tried again. (integer value)
#memcache_pool_dead_retry = 300

# (Optional) Maximum total number of open connections to every
# memcached server. (integer value)
#memcache_pool_maxsize = 10

# (Optional) Socket timeout in seconds for communicating with a
# memcache server. (integer value)
#memcache_pool_socket_timeout = 3

# (Optional) Number of seconds a connection to memcached is held
# unused in the pool before it is closed. (integer value)
#memcache_pool_unused_timeout = 60

# (Optional) Number of seconds that an operation will wait to get a
# memcache client connection from the pool. (integer value)
#memcache_pool_conn_get_timeout = 10

# (Optional) Use the advanced (eventlet safe) memcache client pool.
# The advanced pool will only work under python 2.x. (boolean value)
```

```
#memcache_use_advanced_pool = false

# (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 identity 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

# 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 admin Identity API endpoint. This should specify the
# unversioned root endpoint e.g. https://localhost:35357/ (string
# value)
#identity_uri = <None>

# 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>
```

```
# Service username. (string value)
#admin_user = <None>

# Service user password. (string value)
#admin_password = <None>

# Service tenant name. (string value)
#admin_tenant_name = admin


[matchmaker_redis]

#
# From 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]

#
# From oslo.messaging
#

# Matchmaker ring file (JSON). (string value)
# Deprecated group/name - [DEFAULT]/matchmaker_ringfile
#ringfile = /etc/oslo/matchmaker_ring.json


[notification]

#
# From ceilometer
#

# Acknowledge message when event persistence fails. (boolean value)
# Deprecated group/name - [collector]/ack_on_event_error
#ack_on_event_error = true

# Save event details. (boolean value)
# Deprecated group/name - [collector]/store_events
#store_events = false

# WARNING: Ceilometer historically offered the ability to store events
# as meters. This usage is NOT advised as it can flood the metering
# database and cause performance degradation. This option disables the
# collection of non-metric meters and will be the default behavior in
# Liberty. (boolean value)
#disable_non_metric_meters = false

# Enable workload partitioning, allowing multiple notification agents
```

```
# to be run simultaneously. (boolean value)
#workload_partitioning = false

# Messaging URLs to listen for notifications. Example:
# transport://user:pass@host1:port[,hostN:portN]/virtual_host
# (DEFAULT/transport_url is used if empty) (multi valued)
#messaging_urls =


[oslo_concurrency]

#
# From oslo.concurrency
#

# Enables or disables inter-process locks. (boolean value)
# Deprecated group/name - [DEFAULT]/disable_process_locking
#disable_process_locking = false

# Directory to use for lock files.  For security, the specified
# directory should only be writable by the user running the processes
# that need locking. Defaults to environment variable OSLO_LOCK_PATH.
# If external locks are used, a lock path must be set. (string value)
# Deprecated group/name - [DEFAULT]/lock_path
#lock_path = <None>


[oslo_messaging_amqp]

#
# From oslo.messaging
#

# address prefix used when sending to a specific server (string value)
# Deprecated group/name - [amqp1]/server_request_prefix
#server_request_prefix = exclusive

# address prefix used when broadcasting to all servers (string value)
# Deprecated group/name - [amqp1]/broadcast_prefix
#broadcast_prefix = broadcast

# address prefix when sending to any server in group (string value)
# Deprecated group/name - [amqp1]/group_request_prefix
#group_request_prefix = unicast

# Name for the AMQP container (string value)
# Deprecated group/name - [amqp1]/container_name
#container_name = <None>

# Timeout for inactive connections (in seconds) (integer value)
# Deprecated group/name - [amqp1]/idle_timeout
#idle_timeout = 0

# Debug: dump AMQP frames to stdout (boolean value)
# Deprecated group/name - [amqp1]/trace
#trace = false

# CA certificate PEM file to verify server certificate (string
# value)
# Deprecated group/name - [amqp1]/ssl_ca_file
```

```
#ssl_ca_file =

# Identifying certificate PEM file to present to clients (string
# value)
# Deprecated group/name - [amqp1]/ssl_cert_file
#ssl_cert_file =

# Private key PEM file used to sign cert_file certificate (string
# value)
# Deprecated group/name - [amqp1]/ssl_key_file
#ssl_key_file =

# Password for decrypting ssl_key_file (if encrypted) (string value)
# Deprecated group/name - [amqp1]/ssl_key_password
#ssl_key_password = <None>

# Accept clients using either SSL or plain TCP (boolean value)
# Deprecated group/name - [amqp1]/allow_insecure_clients
#allow_insecure_clients = false


[oslo_messaging_qpid]

#
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
#amqp_auto_delete = false

# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
#rpc_conn_pool_size = 30

# Qpid broker hostname. (string value)
# Deprecated group/name - [DEFAULT]/qpid_hostname
#qpid_hostname = localhost

# Qpid broker port. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_port
#qpid_port = 5672

# Qpid HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/qpid_hosts
#qpid_hosts = $qpid_hostname:$qpid_port

# Username for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_username
#qpid_username =

# Password for Qpid connection. (string value)
# Deprecated group/name - [DEFAULT]/qpid_password
#qpid_password =

# Space separated list of SASL mechanisms to use for auth. (string
```

```
# value)
# Deprecated group/name - [DEFAULT]/qpid_sasl_mechanisms
#qpid_sasl_mechanisms =

# Seconds between connection keepalive heartbeats. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_heartbeat
#qpid_heartbeat = 60

# Transport to use, either 'tcp' or 'ssl'. (string value)
# Deprecated group/name - [DEFAULT]/qpid_protocol
#qpid_protocol = tcp

# Whether to disable the Nagle algorithm. (boolean value)
# Deprecated group/name - [DEFAULT]/qpid_tcp_nodelay
#qpid_tcp_nodelay = true

# The number of prefetched messages held by receiver. (integer value)
# Deprecated group/name - [DEFAULT]/qpid_receiver_capacity
#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)
# Deprecated group/name - [DEFAULT]/qpid_topology_version
#qpid_topology_version = 1


[oslo_messaging_rabbit]

#
# From 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)
# Deprecated group/name - [DEFAULT]/amqp_auto_delete
#amqp_auto_delete = false

# Size of RPC connection pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_conn_pool_size
#rpc_conn_pool_size = 30

# SSL version to use (valid only if SSL enabled). Valid values are
# TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be
# available on some distributions. (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_version
#kombu_ssl_version =

# SSL key file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_keyfile
#kombu_ssl_keyfile =

# SSL cert file (valid only if SSL enabled). (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_certfile
#kombu_ssl_certfile =
```

```
# SSL certification authority file (valid only if SSL enabled).
# (string value)
# Deprecated group/name - [DEFAULT]/kombu_ssl_ca_certs
#kombu_ssl_ca_certs =

# How long to wait before reconnecting in response to an AMQP consumer
# cancel notification. (floating point value)
# Deprecated group/name - [DEFAULT]/kombu_reconnect_delay
#kombu_reconnect_delay = 1.0

# The RabbitMQ broker address where a single node is used. (string
# value)
# Deprecated group/name - [DEFAULT]/rabbit_host
#rabbit_host = localhost

# The RabbitMQ broker port where a single node is used. (integer
# value)
# Deprecated group/name - [DEFAULT]/rabbit_port
#rabbit_port = 5672

# RabbitMQ HA cluster host:port pairs. (list value)
# Deprecated group/name - [DEFAULT]/rabbit_hosts
#rabbit_hosts = $rabbit_host:$rabbit_port

# Connect over SSL for RabbitMQ. (boolean value)
# Deprecated group/name - [DEFAULT]/rabbit_use_ssl
#rabbit_use_ssl = false

# The RabbitMQ userid. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_userid
#rabbit_userid = guest

# The RabbitMQ password. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_password
#rabbit_password = guest

# The RabbitMQ login method. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_login_method
#rabbit_login_method = AMQPLAIN

# The RabbitMQ virtual host. (string value)
# Deprecated group/name - [DEFAULT]/rabbit_virtual_host
#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)
# Deprecated group/name - [DEFAULT]/rabbit_retry_backoff
#rabbit_retry_backoff = 2

# Maximum number of RabbitMQ connection retries. Default is 0
# (infinite retry count). (integer value)
# Deprecated group/name - [DEFAULT]/rabbit_max_retries
#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)
```

```
# Deprecated group/name - [DEFAULT]/rabbit_ha_queues
#rabbit_ha_queues = false

# Number of seconds after which the Rabbit broker is considered down
# if heartbeat's keep-alive fails (0 disables the heartbeat, >0
# enables it. Enabling heartbeats requires kombu>=3.0.7 and
# amqp>=1.4.0). EXPERIMENTAL (integer value)
#heartbeat_timeout_threshold = 0

# How often times during the heartbeat_timeout_threshold we check the
# heartbeat. (integer value)
#heartbeat_rate = 2

# Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
# (boolean value)
# Deprecated group/name - [DEFAULT]/fake_rabbit
#fake_rabbit = false


[oslo_policy]

#
# From oslo.policy
#

# The JSON file that defines policies. (string value)
# Deprecated group/name - [DEFAULT]/policy_file
#policy_file = policy.json

# Default rule. Enforced when a requested rule is not found. (string
# value)
# Deprecated group/name - [DEFAULT]/policy_default_rule
#policy_default_rule = default

# Directories where policy configuration files are stored. They can be
# relative to any directory in the search path defined by the
# config_dir option, or absolute paths. The file defined by
# policy_file must exist for these directories to be searched.
# Missing or empty directories are ignored. (multi valued)
# Deprecated group/name - [DEFAULT]/policy_dirs
#policy_dirs = policy.d


[polling]

#
# From ceilometer
#

# Work-load partitioning group prefix. Use only if you want to run
# multiple polling agents with different config files. For each sub-
# group of the agent pool with the same partitioning_group_prefix a
# disjoint subset of pollsters should be loaded. (string value)
# Deprecated group/name - [central]/partitioning_group_prefix
#partitioning_group_prefix = <None>


[publisher]

#
```

```
# From ceilometer
#

# Secret value for signing messages. Set value empty if signing is not
# required to avoid computational overhead. (string value)
# Deprecated group/name - [DEFAULT]/metering_secret
# Deprecated group/name - [publisher_rpc]/metering_secret
# Deprecated group/name - [publisher]/metering_secret
#telemetry_secret = change this for valid signing


[publisher_notifier]

#
# From ceilometer
#

# The topic that ceilometer uses for metering notifications. (string
# value)
#metering_topic = metering

# The topic that ceilometer uses for event notifications. (string
# value)
#event_topic = event

# The driver that ceilometer uses for metering notifications. (string
# value)
# Deprecated group/name - [DEFAULT]/metering_driver
#telemetry_driver = messagingv2


[publisher_rpc]

#
# From ceilometer
#

# The topic that ceilometer uses for metering messages. (string value)
# Deprecated group/name - [DEFAULT]/metering_topic
#metering_topic = metering


[rgw_admin_credentials]

#
# From ceilometer
#

# Access key for Radosgw Admin. (string value)
#access_key = <None>

# Secret key for Radosgw Admin. (string value)
#secret_key = <None>


[service_credentials]

#
# From ceilometer
#
```

```
# User name to use for OpenStack service access. (string value)
# Deprecated group/name - [DEFAULT]/os_username
#os_username = ceilometer

# Password to use for OpenStack service access. (string value)
# Deprecated group/name - [DEFAULT]/os_password
#os_password = admin

# Tenant ID to use for OpenStack service access. (string value)
# Deprecated group/name - [DEFAULT]/os_tenant_id
#os_tenant_id =

# Tenant name to use for OpenStack service access. (string value)
# Deprecated group/name - [DEFAULT]/os_tenant_name
#os_tenant_name = admin

# Certificate chain for SSL validation. (string value)
#os_cacert = <None>

# Auth URL to use for OpenStack service access. (string value)
# Deprecated group/name - [DEFAULT]/os_auth_url
#os_auth_url = http://localhost:5000/v2.0

# Region name to use for OpenStack service endpoints. (string value)
# Deprecated group/name - [DEFAULT]/os_region_name
#os_region_name = <None>

# Type of endpoint in Identity service catalog to use for
# communication with OpenStack services. (string value)
#os_endpoint_type = publicURL

# Disables X.509 certificate validation when an SSL connection to
# Identity Service is established. (boolean value)
#insecure = false


[service_types]

#
# From ceilometer
#

# Kwapi service type. (string value)
#kwapi = energy

# Glance service type. (string value)
#glance = image

# Neutron service type. (string value)
#neutron = network

# Nova service type. (string value)
#nova = compute

# Radosgw service type. (string value)
#radosgw = object-store

# Swift service type. (string value)
#swift = object-store
```

```
[vmware]

#
# From ceilometer
#

# IP address of the VMware Vsphere host. (string value)
#host_ip =

# Port of the VMware Vsphere host. (integer value)
#host_port = 443

# Username of VMware Vsphere. (string value)
#host_username =

# Password of VMware Vsphere. (string value)
#host_password =

# Number of times a VMware Vsphere API may 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

# Optional vim service WSDL location e.g
# http://<server>/vimService.wsdl. Optional over-ride to default
# location for bug work-arounds. (string value)
#wsdl_location = <None>


[xenapi]

#
# From ceilometer
#

# URL for connection to XenServer/Xen Cloud Platform. (string value)
#connection_url = <None>

# Username for connection to XenServer/Xen Cloud Platform. (string
# value)
#connection_username = root

# Password for connection to XenServer/Xen Cloud Platform. (string
# value)
#connection_password = <None>

# Timeout in seconds for XenAPI login. (integer value)
#login_timeout = 10
```

event_definitions.yaml

The `event_definitions.yaml` file defines how events received from other OpenStack components should be translated to Telemetry events.

This file provides a standard set of events and corresponding traits that may be of interest. This file can be modified to add and drop traits that operators may find useful.

```
---
- 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
- event_type: ['volume.exists', 'volume.create.*', 'volume.delete.*', 'volume.
resize.*', 'volume.attach.*', 'volume.detach.*', 'volume.update.*', 'snapshot.
exists', 'snapshot.create.*', 'snapshot.delete.*', 'snapshot.update.*']
  traits: &cinder_traits
    user_id:
      fields: payload.user_id
    project_id:
      fields: payload.tenant_id
    availability_zone:
      fields: payload.availability_zone
    display_name:
      fields: payload.display_name
    replication_status:
      fields: payload.replication_status
    status:
      fields: payload.status
    created_at:
      fields: payload.created_at
- event_type: ['volume.exists', 'volume.create.*', 'volume.delete.*', 'volume.
resize.*', 'volume.attach.*', 'volume.detach.*', 'volume.update.*']
  traits:
    <<: *cinder_traits
    resource_id:
      fields: payload.volume_id
    host:
      fields: payload.host
    size:
      fields: payload.size
    type:
      fields: payload.volume_type
    replication_status:
      fields: payload.replication_status
- event_type: ['snapshot.exists', 'snapshot.create.*', 'snapshot.delete.*',
 'snapshot.update.*']
  traits:
    <<: *cinder_traits
    resource_id:
      fields: payload.snapshot_id
    volume_id:
      fields: payload.volume_id
- event_type: ['image.update', 'image.upload', 'image.delete']
  traits: &glance_crud
    project_id:
      fields: payload.owner
    resource_id:
      fields: payload.id
    name:
      fields: payload.name
    status:
      fields: payload.status
    created_at:
      fields: payload.created_at
    user_id:
      fields: payload.owner
    deleted_at:
```

```
      fields: payload.deleted_at
    size:
      fields: payload.size
- event_type: image.send
  traits: &glance_send
    receiver_project:
      fields: payload.receiver_tenant_id
    receiver_user:
      fields: payload.receiver_user_id
    user_id:
      fields: payload.owner_id
    image_id:
      fields: payload.image_id
    destination_ip:
      fields: payload.destination_ip
    bytes_sent:
      fields: payload.bytes_sent
- event_type: orchestration.stack.*
  traits: &orchestration_crud
    project_id:
      fields: payload.tenant_id
    user_id:
      fields: ['_context_trustor_user_id', '_context_user_id']
    resource_id:
      fields: payload.stack_identity
- event_type: sahara.cluster.*
  traits: &sahara_crud
    project_id:
      fields: payload.project_id
    user_id:
      fields: _context_user_id
    resource_id:
      fields: payload.cluster_id
- event_type: ['identity.user.*', 'identity.project.*', 'identity.group.*',
 'identity.role.*', 'identity.OS-TRUST:trust.*',
               'identity.region.*', 'identity.service.*', 'identity.endpoint.
*', 'identity.policy.*']
  traits: &identity_crud
    resource_id:
      fields: payload.resource_info
    initiator_id:
      fields: payload.initiator.id
    project_id:
      fields: payload.initiator.project_id
    domain_id:
      fields: payload.initiator.domain_id
- event_type: identity.role_assignment.*
  traits: &identity_role_assignment
    role:
      fields: payload.role
    group:
      fields: payload.group
    domain:
      fields: payload.domain
    user:
      fields: payload.user
    project:
      fields: payload.project
- event_type: identity.authenticate
  traits: &identity_authenticate
```

```
    typeURI:
      fields: payload.typeURI
    id:
      fields: payload.id
    action:
      fields: payload.action
    eventType:
      fields: payload.eventType
    eventTime:
      fields: payload.eventTime
    outcome:
      fields: payload.outcome
    initiator_typeURI:
      fields: payload.initiator.typeURI
    initiator_id:
      fields: payload.initiator.id
    initiator_name:
      fields: payload.initiator.name
    initiator_host_agent:
      fields: payload.initiator.host.agent
    initiator_host_addr:
      fields: payload.initiator.host.address
    target_typeURI:
      fields: payload.target.typeURI
    target_id:
      fields: payload.target.id
    observer_typeURI:
      fields: payload.observer.typeURI
    observer_id:
      fields: payload.observer.id
- event_type: objectstore.http.request
  traits: &objectstore_request
    typeURI:
      fields: payload.typeURI
    id:
      fields: payload.id
    action:
      fields: payload.action
    eventType:
      fields: payload.eventType
    eventTime:
      fields: payload.eventTime
    outcome:
      fields: payload.outcome
    initiator_typeURI:
      fields: payload.initiator.typeURI
    initiator_id:
      fields: payload.initiator.id
    initiator_project_id:
      fields: payload.initiator.project_id
    target_typeURI:
      fields: payload.target.typeURI
    target_id:
      fields: payload.target.id
    target_action:
      fields: payload.target.action
    target_metadata_path:
      fields: payload.target.metadata.path
    target_metadata_version:
      fields: payload.target.metadata.version
```

```
    target_metadata_container:
      fields: payload.target.metadata.container
    target_metadata_object:
      fields: payload.target.metadata.object
    observer_id:
      fields: payload.observer.id
- event_type: magnetodb.table.*
  traits: &kv_store
    resource_id:
      fields: payload.table_uuid
    user_id:
      fields: _context_user_id
    project_id:
      fields: _context_tenant
- event_type: ['network.*', 'subnet.*', 'port.*', 'router.*', 'floatingip.
*', 'pool.*', 'vip.*', 'member.*', 'health_monitor.*', 'firewall.*',
 'firewall_policy.*', 'firewall_rule.*', 'vpnservice.*', 'ipsecpolicy.*',
 'ikepolicy.*', 'ipsec_site_connection.*']
  traits: &network_traits
    user_id:
      fields: _context_user_id
    project_id:
      fields: _context_tenant_id
- event_type: network.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.network.id', 'payload.id']
- event_type: subnet.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.subnet.id', 'payload.id']
- event_type: port.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.port.id', 'payload.id']
- event_type: router.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.router.id', 'payload.id']
- event_type: floatingip.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.floatingip.id', 'payload.id']
- event_type: pool.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.pool.id', 'payload.id']
- event_type: vip.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.vip.id', 'payload.id']
- event_type: member.*
  traits:
```

```
    <<: *network_traits
    resource_id:
      fields: ['payload.member.id', 'payload.id']
- event_type: health_monitor.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.health_monitor.id', 'payload.id']
- event_type: firewall.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.firewall.id', 'payload.id']
- event_type: firewall_policy.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.firewall_policy.id', 'payload.id']
- event_type: firewall_rule.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.firewall_rule.id', 'payload.id']
- event_type: vpnservice.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.vpnservice.id', 'payload.id']
- event_type: ipsecpolicy.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.ipsecpolicy.id', 'payload.id']
- event_type: ikepolicy.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.ikepolicy.id', 'payload.id']
- event_type: ipsec_site_connection.*
  traits:
    <<: *network_traits
    resource_id:
      fields: ['payload.ipsec_site_connection.id', 'payload.id']
- event_type: '*http.*'
  traits: &http_audit
    typeURI:
      fields: payload.typeURI
    eventType:
      fields: payload.eventType
    action:
      fields: payload.action
    outcome:
      fields: payload.outcome
    id:
      fields: payload.id
    eventTime:
      fields: payload.eventTime
    requestPath:
      fields: payload.requestPath
    observer_id:
```

```
      fields: payload.observer.id
    target_id:
      fields: payload.target.id
    target_typeURI:
      fields: payload.target.typeURI
    target_name:
      fields: payload.target.name
    initiator_typeURI:
      fields: payload.initiator.typeURI
    initiator_id:
      fields: payload.initiator.id
    initiator_name:
      fields: payload.initiator.name
    initiator_host_address:
      fields: payload.initiator.host.address
- event_type: '*http.response'
  traits:
    <<: *http_audit
    reason_code:
      fields: payload.reason.reasonCode
```

pipeline.yaml

Pipelines describe a coupling between sources of samples and the corresponding sinks for transformation and publication of the data. They are defined in the `pipeline.yaml` file.

This file can be modified to adjust polling intervals and the samples generated by the Telemetry module

```
---
sources:
    - name: meter_source
      interval: 600
      meters:
          - "*"
      sinks:
          - meter_sink
    - name: cpu_source
      interval: 600
      meters:
          - "cpu"
      sinks:
          - cpu_sink
    - name: disk_source
      interval: 600
      meters:
          - "disk.read.bytes"
          - "disk.read.requests"
          - "disk.write.bytes"
          - "disk.write.requests"
          - "disk.device.read.bytes"
          - "disk.device.read.requests"
          - "disk.device.write.bytes"
          - "disk.device.write.requests"
      sinks:
          - disk_sink
    - name: network_source
      interval: 600
      meters:
```

```
          - "network.incoming.bytes"
          - "network.incoming.packets"
          - "network.outgoing.bytes"
          - "network.outgoing.packets"
      sinks:
          - network_sink
sinks:
    - name: meter_sink
      transformers:
      publishers:
          - notifier://
    - name: cpu_sink
      transformers:
          - name: "rate_of_change"
            parameters:
                target:
                    name: "cpu_util"
                    unit: "%"
                    type: "gauge"
                    scale: "100.0 / (10**9 * (resource_metadata.cpu_number or
 1))"
      publishers:
          - notifier://
    - name: disk_sink
      transformers:
          - name: "rate_of_change"
            parameters:
                source:
                    map_from:
                        name: "(disk\\.device|disk)\\.(read|write)\\.(bytes|
requests)"
                        unit: "(B|request)"
                target:
                    map_to:
                        name: "\\1.\\2.\\3.rate"
                        unit: "\\1/s"
                    type: "gauge"
      publishers:
          - notifier://
    - name: network_sink
      transformers:
          - name: "rate_of_change"
            parameters:
                source:
                   map_from:
                       name: "network\\.(incoming|outgoing)\\.(bytes|packets)"
                       unit: "(B|packet)"
                target:
                    map_to:
                        name: "network.\\1.\\2.rate"
                        unit: "\\1/s"
                    type: "gauge"
      publishers:
          - notifier://
```

event_pipeline.yaml

Event pipelines describe a coupling between notification event_types and the corresponding sinks for publication of the event data. They are defined in the `event_pipeline.yaml` file.

This file can be modified to adjust which notifications to capture and the and where to publish the events.

```
---
sources:
    - name: event_source
      events:
          - "*"
      sinks:
          - event_sink
sinks:
    - name: event_sink
      transformers:
      triggers:
      publishers:
          - direct://
```

policy.json

The `policy.json` file defines additional access controls that apply to the Telemetry service.

```
{
    "context_is_admin": "role:admin",
    "context_is_project": "project_id:%(target.project_id)s",
    "context_is_owner": "user_id:%(target.user_id)s",
    "segregation": "rule:context_is_admin",
    "default": ""
}
```

New, updated and deprecated options in Kilo for Telemetry

Table 12.35. New options

Option = default value	(Type) Help string
[DEFAULT] api_workers = 1	(IntOpt) Number of workers for Ceilometer API server.
[DEFAULT] event_pipeline_cfg_file = event_pipeline.yaml	(StrOpt) Configuration file for event pipeline definition.
[DEFAULT] magnetodb_control_exchange = magnetodb	(StrOpt) Exchange name for Magnetodb notifications.
[DEFAULT] polling_namespaces = ['compute', 'central']	(MultiChoicesOpt) Polling namespace(s) to be used while resource polling
[DEFAULT] pollster_list = []	(MultiChoicesOpt) List of pollsters (or wildcard templates) to be used while polling
[DEFAULT] reserved_metadata_keys =	(ListOpt) List of metadata keys reserved for metering use. And these keys are additional to the ones included in the namespace.

Option = default value	(Type) Help string
[DEFAULT] shuffle_time_before_polling_task = 0	(IntOpt) To reduce large requests at same time to Nova or other components from different compute agents, shuffle start time of polling task.
[DEFAULT] sql_expire_samples_only = False	(BoolOpt) Indicates if expirer expires only samples. If set true, expired samples will be deleted, but residual resource and meter definition data will remain.
[DEFAULT] swift_control_exchange = swift	(StrOpt) Exchange name for Swift notifications.
[DEFAULT] zaqar_control_exchange = zaqar	(StrOpt) Exchange name for Messaging service notifications.
[alarms] gnocchi_url = http://localhost:8041	(StrOpt) URL to Gnocchi.
[collector] requeue_event_on_dispatcher_error = False	(BoolOpt) Requeue the event on the collector event queue when the collector fails to dispatch it.
[coordination] check_watchers = 10.0	(FloatOpt) Number of seconds between checks to see if group membership has changed
[database] db2nosql_resource_id_maxlen = 512	(IntOpt) The max length of resources id in DB2 nosql, the value should be larger than len(hostname) * 2 as compute node's resource id is <hostname>_<nodename>.
[database] event_connection = None	(StrOpt) The connection string used to connect to the event database. (if unset, connection is used)
[database] event_time_to_live = -1	(IntOpt) Number of seconds that events are kept in the database for (<= 0 means forever).
[database] metering_time_to_live = -1	(IntOpt) Number of seconds that samples are kept in the database for (<= 0 means forever).
[database] mongodb_replica_set =	(StrOpt) The name of the replica set which is used to connect to MongoDB database. If it is set, MongoReplicaSetClient will be used instead of MongoClient.
[dispatcher_http] cadf_only = False	(BoolOpt) The flag that indicates if only cadf message should be posted. If false, all meters will be posted.
[dispatcher_http] event_target = None	(StrOpt) The target for event data where the http request will be sent to. If this is not set, it will default to same as Sample target.
[dispatcher_http] target =	(StrOpt) The target where the http request will be sent. If this is not set, no data will be posted. For example: target = http://hostname:1234/path
[dispatcher_http] timeout = 5	(IntOpt) The max time in seconds to wait for a request to timeout.
[event] store_raw = []	(MultiStrOpt) Store the raw notification for select priority levels (info and/or error). By default, raw details are not captured.
[ipmi] polling_retry = 3	(IntOpt) Tolerance of IPMI/NM polling failures before disable this pollster. Negative indicates retrying forever.
[notification] disable_non_metric_meters = False	(BoolOpt) WARNING: Ceilometer historically offered the ability to store events as meters. This usage is NOT advised as it can flood the metering database and cause performance degradation. This option disables the collection of non-metric meters and will be the default behavior in Liberty.
[notification] workload_partitioning = False	(BoolOpt) Enable workload partitioning, allowing multiple notification agents to be run simultaneously.
[oslo_concurrency] disable_process_locking = False	(BoolOpt) Enables or disables inter-process locks.
[oslo_concurrency] lock_path = None	(StrOpt) Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.

Option = default value	(Type) Help string
[oslo_messaging_amqp] allow_insecure_clients = False	(BoolOpt) Accept clients using either SSL or plain TCP
[oslo_messaging_amqp] broadcast_prefix = broadcast	(StrOpt) address prefix used when broadcasting to all servers
[oslo_messaging_amqp] container_name = None	(StrOpt) Name for the AMQP container
[oslo_messaging_amqp] group_request_prefix = unicast	(StrOpt) address prefix when sending to any server in group
[oslo_messaging_amqp] idle_timeout = 0	(IntOpt) Timeout for inactive connections (in seconds)
[oslo_messaging_amqp] server_request_prefix = exclusive	(StrOpt) address prefix used when sending to a specific server
[oslo_messaging_amqp] ssl_ca_file =	(StrOpt) CA certificate PEM file to verify server certificate
[oslo_messaging_amqp] ssl_cert_file =	(StrOpt) Identifying certificate PEM file to present to clients
[oslo_messaging_amqp] ssl_key_file =	(StrOpt) Private key PEM file used to sign cert_file certificate
[oslo_messaging_amqp] ssl_key_password = None	(StrOpt) Password for decrypting ssl_key_file (if encrypted)
[oslo_messaging_amqp] trace = False	(BoolOpt) Debug: dump AMQP frames to stdout
[oslo_messaging_qpid] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_qpid] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_qpid] qpid_heartbeat = 60	(IntOpt) Seconds between connection keepalive heartbeats.
[oslo_messaging_qpid] qpid_hostname = localhost	(StrOpt) Qpid broker hostname.
[oslo_messaging_qpid] qpid_hosts = $qpid_hostname:$qpid_port	(ListOpt) Qpid HA cluster host:port pairs.
[oslo_messaging_qpid] qpid_password =	(StrOpt) Password for Qpid connection.
[oslo_messaging_qpid] qpid_port = 5672	(IntOpt) Qpid broker port.
[oslo_messaging_qpid] qpid_protocol = tcp	(StrOpt) Transport to use, either 'tcp' or 'ssl'.
[oslo_messaging_qpid] qpid_receiver_capacity = 1	(IntOpt) The number of prefetched messages held by receiver.
[oslo_messaging_qpid] qpid_sasl_mechanisms =	(StrOpt) Space separated list of SASL mechanisms to use for auth.
[oslo_messaging_qpid] qpid_tcp_nodelay = True	(BoolOpt) Whether to disable the Nagle algorithm.
[oslo_messaging_qpid] 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.
[oslo_messaging_qpid] qpid_username =	(StrOpt) Username for Qpid connection.
[oslo_messaging_qpid] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_messaging_rabbit] amqp_auto_delete = False	(BoolOpt) Auto-delete queues in AMQP.
[oslo_messaging_rabbit] amqp_durable_queues = False	(BoolOpt) Use durable queues in AMQP.
[oslo_messaging_rabbit] fake_rabbit = False	(BoolOpt) Deprecated, use rpc_backend=kombu+memory or rpc_backend=fake
[oslo_messaging_rabbit] heartbeat_rate = 2	(IntOpt) How often times during the heartbeat_timeout_threshold we check the heartbeat.
[oslo_messaging_rabbit] heartbeat_timeout_threshold = 0	(IntOpt) Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables the heartbeat, >0 enables it. Enabling heartbeats requires kombu>=3.0.7 and amqp>=1.4.0). EXPERIMENTAL

Option = default value	(Type) Help string
[oslo_messaging_rabbit] kombu_reconnect_delay = 1.0	(FloatOpt) How long to wait before reconnecting in response to an AMQP consumer cancel notification.
[oslo_messaging_rabbit] kombu_ssl_ca_certs =	(StrOpt) SSL certification authority file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_certfile =	(StrOpt) SSL cert file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_keyfile =	(StrOpt) SSL key file (valid only if SSL enabled).
[oslo_messaging_rabbit] kombu_ssl_version =	(StrOpt) SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.
[oslo_messaging_rabbit] 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.
[oslo_messaging_rabbit] rabbit_host = localhost	(StrOpt) The RabbitMQ broker address where a single node is used.
[oslo_messaging_rabbit] rabbit_hosts = $rabbit_host:$rabbit_port	(ListOpt) RabbitMQ HA cluster host:port pairs.
[oslo_messaging_rabbit] rabbit_login_method = AMQPLAIN	(StrOpt) The RabbitMQ login method.
[oslo_messaging_rabbit] rabbit_max_retries = 0	(IntOpt) Maximum number of RabbitMQ connection retries. Default is 0 (infinite retry count).
[oslo_messaging_rabbit] rabbit_password = guest	(StrOpt) The RabbitMQ password.
[oslo_messaging_rabbit] rabbit_port = 5672	(IntOpt) The RabbitMQ broker port where a single node is used.
[oslo_messaging_rabbit] rabbit_retry_backoff = 2	(IntOpt) How long to backoff for between retries when connecting to RabbitMQ.
[oslo_messaging_rabbit] rabbit_retry_interval = 1	(IntOpt) How frequently to retry connecting with RabbitMQ.
[oslo_messaging_rabbit] rabbit_use_ssl = False	(BoolOpt) Connect over SSL for RabbitMQ.
[oslo_messaging_rabbit] rabbit_userid = guest	(StrOpt) The RabbitMQ userid.
[oslo_messaging_rabbit] rabbit_virtual_host = /	(StrOpt) The RabbitMQ virtual host.
[oslo_messaging_rabbit] rpc_conn_pool_size = 30	(IntOpt) Size of RPC connection pool.
[oslo_middleware] max_request_body_size = 114688	(IntOpt) The maximum body size for each request, in bytes.
[oslo_policy] policy_default_rule = default	(StrOpt) Default rule. Enforced when a requested rule is not found.
[oslo_policy] policy_dirs = ['policy.d']	(MultiStrOpt) Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the config_dir option, or absolute paths. The file defined by policy_file must exist for these directories to be searched. Missing or empty directories are ignored.
[oslo_policy] policy_file = policy.json	(StrOpt) The JSON file that defines policies.
[polling] partitioning_group_prefix = None	(StrOpt) Work-load partitioning group prefix. Use only if you want to run multiple polling agents with different config files. For each sub-group of the agent pool with the same partitioning_group_prefix a disjoint subset of pollsters should be loaded.
[publisher] telemetry_secret = change this for valid signing	(StrOpt) Secret value for signing messages. Set value empty if signing is not required to avoid computational overhead.
[publisher_notifier] event_topic = event	(StrOpt) The topic that ceilometer uses for event notifications.

Option = default value	(Type) Help string
[publisher_notifier] telemetry_driver = messagingv2	(StrOpt) The driver that ceilometer uses for metering notifications.
[rgw_admin_credentials] access_key = None	(StrOpt) Access key for Radosgw Admin.
[rgw_admin_credentials] secret_key = None	(StrOpt) Secret key for Radosgw Admin.
[service_types] radosgw = object-store	(StrOpt) Radosgw service type.
[vmware] host_port = 443	(IntOpt) Port of the VMware Vsphere host.

Table 12.36. New default values

Option	Previous default value	New default value
[DEFAULT] rpc_zmq_matchmaker	oslo.messaging._drivers.matchmaker.MatchMakerLocal	oslo.messaging._drivers.matchmaker.MatchMakerLocalhost

Table 12.37. Deprecated options

Deprecated option	New Option
[alarm] evaluation_service	None
[DEFAULT] log_format	None
[database] time_to_live	[database] metering_time_to_live
[DEFAULT] database_connection	None
[publisher] metering_secret	[publisher] telemetry_secret
[alarm] partition_rpc_topic	None
[DEFAULT] use_syslog	None

Appendix A. The policy.json file

Table of Contents

Each OpenStack service, Identity, Compute, Networking and so on, has its own role-based access policies. They determine which user can access which objects in which way, and are defined in the service's `policy.json` file.

Whenever an API call to an OpenStack service is made, the service's policy engine uses the appropriate policy definitions to determine if the call can be accepted. Any changes to `policy.json` are effective immediately, which allows new policies to be implemented while the service is running.

A `policy.json` file is a text file in JSON (Javascript Object Notation) format. Each policy is defined by a one-line statement in the form `"<target>" : "<rule>"`.

The policy target, also named "action", represents an API call like "start an instance" or "attach a volume".

Action names are usually qualified. Example: OpenStack Compute features API calls to list instances, volumes and networks. In `/etc/nova/policy.json`, these APIs are represented by `compute:get_all`, `volume:get_all` and `network:get_all`, respectively.

The mapping between API calls and actions is not generally documented.

The policy rule determines under which circumstances the API call is permitted. Usually this involves the user who makes the call (hereafter named the "API user") and often the object on which the API call operates. A typical rule checks if the API user is the object's owner.

Modifying the policy

While recipes for editing `policy.json` files are found on blogs, modifying the policy can have unexpected side effects and is not encouraged.

Examples

A simple rule might look like this:

```
"compute:get_all" : ""
```

The target is `"compute:get_all"`, the "list all instances" API of the Compute service. The rule is an empty string meaning "always". This policy allows anybody to list instances.

You can also decline permission to use an API:

```
"compute:shelve": "!"
```

The exclamation mark stands for "never" or "nobody", which effectively disables the Compute API "shelve an instance".

Many APIs can only be called by admin users. This can be expressed by the rule `"role:admin"`. The following policy ensures that only administrators can create new users in the Identity database:

```
"identity:create_user" : "role:admin"
```

You can limit APIs to any role. For example, the Orchestration service defines a role named `heat_stack_user`. Whoever has this role isn't allowed to create stacks:

```
"stacks:create": "not role:heat_stack_user"
```

This rule makes use of the boolean operator `not`. More complex rules can be built using operators `and`, `or` and parentheses.

You can define aliases for rules:

```
"deny_stack_user": "not role:heat_stack_user"
```

The policy engine understands that `"deny_stack_user"` is not an API and consequently interprets it as an alias. The stack creation policy above can then be written as:

```
"stacks:create": "rule:deny_stack_user"
```

This is taken verbatim from `/etc/heat/policy.json`.

Rules can compare API attributes to object attributes. For example:

```
"compute:start" : "user_id:%(user_id)s"
```

states that only the owner of an instance can start it up. The `user_id` string before the colon is an API attribute, namely the user ID of the API user. It is compared with the user ID of the object (in this case, an instance); more precisely, it is compared with the `user_id` field of that object in the database. If the two values are equal, permission is granted.

An admin user always has permission to call APIs. This is how `/etc/keystone/policy.json` makes this policy explicit:

```
"admin_required": "role:admin or is_admin:1",
"owner" : "user_id:%(user_id)s",
"admin_or_owner": "rule:admin_required or rule:owner",
"identity:change_password": "rule:admin_or_owner"
```

The first line defines an alias for "user is an admin user". The `is_admin` flag is only used when setting up the Identity service for the first time. It indicates that the user has admin privileges granted by the service token (*`--os-token`* parameter of the **keystone** command line client).

The second line creates an alias for "user owns the object" by comparing the API's user ID with the object's user ID.

Line 3 defines a third alias `admin_or_owner`, combining the two first aliases with the Boolean operator `or`.

Line 4 sets up the policy that a password can only be modified by its owner or an admin user.

As a final example, let's examine a more complex rule:

```
"identity:ec2_delete_credential": "rule:admin_required or
            (rule:owner and user_id:%(target.credential.user_id)s)"
```

This rule determines who can use the Identity API "delete EC2 credential". Here, boolean operators and parentheses combine three simpler rules. `admin_required` and `owner` are the same aliases as in the previous example. `user_id:%(target.credential.user_id)s` compares the API user with the user ID of the credential object associated with the target.

Syntax

A `policy.json` file consists of policies and aliases of the form `target:rule` or `alias:definition`, separated by commas and enclosed in curly braces:

```
{
      "alias 1" : "definition 1",
      "alias 2" : "definition 2",
      ...
      "target 1" : "rule 1",
      "target 2" : "rule 2",
      ....
}
```

Targets are APIs and are written `"service:API"` or simply `"API"`. For example, `"compute:create"` or `"add_image"`.

Rules determine whether the API call is allowed.

Rules can be:

- always true. The action is always permitted. This can be written as `""` (empty string), `[]`, or `"@"`.
- always false. The action is never permitted. Written as `"!"`.
- a special check
- a comparison of two values
- boolean expressions based on simpler rules

Special checks are

- `<role>:<role name>`, a test whether the API credentials contain this role.
- `<rule>:<rule name>`, the definition of an alias.
- `http:<target URL>`, which delegates the check to a remote server. The API is authorized when the server returns True.

Developers can define additional special checks.

Two values are compared in the following way:

```
"value1 : value2"
```

Possible values are

- constants: Strings, numbers, `true`, `false`
- API attributes
- target object attributes
- the flag `is_admin`

API attributes can be `project_id`, `user_id` or `domain_id`.

Target object attributes are fields from the object description in the database. For example in the case of the `"compute:start"` API, the object is the instance to be started. The policy for starting instances could use the `%(project_id)s`attribute, that is the project that owns the instance. The trailing s indicates this is a string.

`is_admin` indicates that administrative privileges are granted via the admin token mechanism (the *`--os-token`* option of the **keystone** command). The admin token allows initialisation of the identity database before the admin role exists.

The alias construct exists for convenience. An alias is short name for a complex or hard to understand rule. It is defined in the same way as a policy:

```
alias name : alias definition
```

Once an alias is defined, use the `rule` keyword to use it in a policy rule.

Older syntax

You may encounter older `policy.json` files that feature a different syntax, where JavaScript arrays are used instead of boolean operators. For example, the EC2 credentials rule above would have been written as follows:

```
"identity:ec2_delete_credential": [ [ "rule:admin_required ],
            [ "rule:owner", "user_id:%(target.credential.user_id)s)" ] ]
```

The rule is an array of arrays. The innermost arrays are or'ed together, whereas elements inside the innermost arrays are and'ed.

While the old syntax is still supported, we recommend using the newer, more intuitive syntax.

Appendix B. 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 B.1. Default ports that OpenStack components use

OpenStack service	Default ports	Port type
Block Storage (`cinder`)	8776	publicurl and adminurl
Compute (`nova`) endpoints	8774	publicurl and adminurl
Compute API (`nova-api`)	8773, 8775	
Compute ports for access to virtual machine consoles	5900-5999	
Compute VNC proxy for browsers (`openstack-nova-novncproxy`)	6080	
Compute VNC proxy for traditional VNC clients (`openstack-nova-xvpvncproxy`)	6081	
Proxy port for HTML5 console used by Compute service	6082	
Data processing service (`sahara`) endpoint	8386	publicurl and adminurl
Identity service (`keystone`) administrative endpoint	35357	adminurl
Identity service public endpoint	5000	publicurl
Image service (`glance`) API	9292	publicurl and adminurl
Image service registry	9191	
Networking (`neutron`)	9696	publicurl and adminurl
Object Storage (`swift`)	6000, 6001, 6002	
Orchestration (`heat`) endpoint	8004	publicurl and adminurl
Orchestration AWS CloudFormation-compatible API (`openstack-heat-api-cfn`)	8000	
Orchestration AWS CloudWatch-compatible API (`openstack-heat-api-cloudwatch`)	8003	
Telemetry (`ceilometer`)	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 B.2. Default ports that secondary services related to OpenStack components use

Service	Default port	Used by
HTTP	80	OpenStack dashboard (`Horizon`) when it is not configured to use secure access.
HTTP alternate	8080	OpenStack Object Storage (`swift`) service.
HTTPS	443	Any OpenStack service that is enabled for SSL, especially secure-access dashboard.
rsync	873	OpenStack Object Storage. Required.

Service	Default port	Used by
iSCSI target	3260	OpenStack Block Storage. Required.
MySQL database service	3306	Most OpenStack components.
Message Broker (AMQP traffic)	5672	OpenStack Block Storage, Networking, Orchestration, and Compute.

On some deployments, the default port used by a service may fall within the defined local port range of a host. To check a host's local port range:

```
$ sysctl -a | grep ip_local_port_range
```

If a service's default port falls within this range, run the following program to check if the port has already been assigned to another application:

```
$ lsof -i :PORT
```

Configure the service to use a different port if the default port is already being used by another application.

Appendix C. Community support

Table of Contents

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 openSUSE 13.2 and SUSE Linux Enterprise Server 12*
- *Installation Guide for Red Hat Enterprise Linux 7, CentOS 7, and Fedora 21*
- *Installation Guide for Ubuntu 14.04 (LTS)*

The following books explain how to configure and run an OpenStack cloud:

- *Architecture Design Guide*
- *Cloud Administrator Guide*
- *Configuration Reference*
- *Operations Guide*
- *Networking 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)

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 https://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 https://wiki.openstack.org/wiki/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 OpenStack Compute, 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 service (ironic)
- Bugs: Data processing service (sahara)
- Bugs: Database service (trove)
- Bugs: Orchestration (heat)
- Bugs: Telemetry (ceilometer)
- Bugs: Message Service (zaqar)
- Bugs: OpenStack API Documentation (developer.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 https://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:** https://wiki.debian.org/OpenStack
- **CentOS, Fedora, and Red Hat Enterprise Linux:** https://www.rdoproject.org/
- **openSUSE and SUSE Linux Enterprise Server:** https://en.opensuse.org/Portal:OpenStack
- **Ubuntu:** https://wiki.ubuntu.com/ServerTeam/CloudArchive